



Universidad
Carlos III de Madrid

Departamento de Ingeniería de Sistemas y Automática.

TRABAJO DE FIN DE GRADO

DISEÑO DE UN ROBOT MÓVIL PARA ESTUDIO DE LA OCUPACIÓN EN HABITACIONES.

Autor: Rubén Cabalgante Ballesta.

Tutor: Jonathan Crespo Herrero.

Leganés, 5 de Marzo de 2015.

Título:

Autor:

Director:

EL TRIBUNAL

Presidente:

Vocal:

Secretario:

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Quiero dedicar este proyecto a mi familia que me han apoyado durante toda la carrera y durante tantas horas dedicadas a este proyecto pero especialmente dedicado a mi yayo que estaría orgulloso y le hubiera gustado verme acabar la carrera.

ÍNDICE GENERAL

RESUMEN	15
ABSTRACT.	16
1. INTRODUCCIÓN	17
1.1. Motivación.	17
1.2. Objetivos.	18
2. ESTADO DEL ARTE.	19
2.1. Detección de rostros.	19
2.1.1. Técnicas.	20
2.1.2. Aplicaciones.	23
2.2. Monitorización de entornos.	27
2.2.1. Robots móviles monitorizados.	27
2.2.2. Sistemas robóticos de monitorización	32
2.3. Empleo de Raspberry Pi y Arduino en la robótica.	33
2.3.1. Aplicaciones monitorizadas.	35
2.3.2. Robots móviles.	37
2.4. ROS.	40
3. SISTEMA DESARROLLADO.	43
3.1. Parte 1: Robot móvil.	46
3.1.1. Chasis del robot.	46
3.1.2. Etapa de potencia (L298N).	48
3.1.3. Sensores ultrasonido (HC-SR04).	49
3.1.4. Arduino UNO.	52
3.1.5. Arduino Sensor Shield.	53
3.2. Parte 2: Detección de personas y habitación.	54
3.2.1. Raspberry Pi.	54
3.2.2. Sensor infrarrojo (Keyes IR-01).	55
3.2.3. Adaptador Wifi (Belkin FD7050).	57
3.2.4. Cámara (EyeToy Namtai).	57

3.3. Bateria.	58
4. DESARROLLO DEL SISTEMA.	61
4.1. Parte 1: Robot móvil.	61
4.1.1. Sensores de ultrasonido.	62
4.1.2. Control de motores.	64
4.1.3. Flujograma de la programación del robot móvil.	66
4.2. Parte 2: Detección personas y habitaciones.	67
4.2.1. Instalación del Sistema Operativo en la Raspberry Pi.	67
4.2.2. Configuración del control remoto de Raspberry Pi.	68
4.2.3. Configuración adaptador Wifi Belkin.	68
4.2.4. Instalación librerías OpenCV.	68
4.2.5. Librerías GPIO de Raspberry Pi.	69
4.2.6. Configuración del PC.	72
4.2.7. Detección de personas.	72
4.2.8. Utilización de ROS para la monitorización.	77
4.2.9. Instalación de ROS.	77
4.2.10. Configuración variables de entorno de ROS.	82
4.2.11. Iniciaización del sistema de monitorización Raspberry Pi-PC.	82
5. EXPERIMENTOS.	85
5.1. Altura de la cámara.	85
5.2. Cambios de iluminación.	89
5.3. Detección de 1 persona en distintas habitaciones.	94
5.4. Número máximo de rostros detectables.	97
5.5. Funcionamiento del Sistema de Detección desarrollado en otro robot.	98
6. CONCLUSIONES Y TRABAJOS FUTUROS.	101
6.1. Conclusiones.	101
6.2. Trabajos futuros.	103
REFERENCIAS.	105

Índice de Figuras.

Figura 2.1: Imágenes de la media ideal de delincuente y de hombre honrado.	20
Figura 2.2: Características usadas en clasificadores HAAR.	22
Figura 2.3: Funcionamiento HAAR Cascade Classifier para considerar rostro o no.	22
Figura 2.4: Visualización de la app SceneTap.	25
Figura 2.5: Robot 510 PackBot.	30
Figura 2.6: Robot aspirador.	31
Figura 2.7: Raspberry Pi.	34
Figura 2.8: Pines GPIO de la Raspberry Pi.	34
Figura 2.9: Arduino modelo UNO.	35
Figura 2.10: Sistema propuesto por Priyanga y Ramanan.	36
Figura 2.11: Robots móviles desarrollados en el libro de Margolis. . . .	38
Figura 2.12: Robot Móvil propuesto en el libro de Grimmett.	39
Figura 2.13: Aplicación monitorización y teleoperación con robot. . . .	41
Figura 2.14: En las imágenes se pueden ver los resultados de las detecciones.	42
Figura 2.15: Aplicación de visualización y monitorización de los dispositivos.	42
Figura 3.1: Diagrama de casos de uso entre el usuario y el sistema desarrollado.	43
Figura 3.2: Diagrama de clases del sistema propuesto.	44
Figura 3.3: Robot móvil creado con sistema de detección de rostros. . .	45
Figura 3.4: Chasis Robot (Vista boca abajo).	46
Figura 3.5: Esquema medidas placas del Chasis.	47

Figura 3.6: Ruedas del robot.	47
Figura 3.7: Placa puente H L298N.	48
Figura 3.8: Esquema de colocación del L298N en el chasis.	48
Figura 3.9: Sensor ultrasonidos HC-SR04.	49
Figura 3.10: Funcionamiento sensor de ultrasonidos.	50
Figura 3.11: Esquema de señales enviadas y recibidas por el sensor. . .	50
Figura 3.12: Esquema colocación sensores ultrasonidos.	51
Figura 3.13: Esquema colocación Arduino UNO.	53
Figura 3.14: Sensor Shield de Arduino.	53
Figura 3.15: Esquema colocación Raspberry Pi en placa superior del Chasis.	55
Figura 3.16: Sensor infrarrojos Keyes IR-01.	56
Figura 3.17: Esquema colocación sensor infrarrojo.	56
Figura 3.18: Adaptador Wifi Belkin F5D7050.	57
Figura 3.19: Cámara EyeToy Namtai.	58
Figura 3.20: Colocación cámara sobre bobina de DVDs en robot.	58
Figura 3.21: Batería RAVPower de 10.400mAh con salidas a 5V de 1A y 2A.	59
Figura 4.1: Conexionado de los sensores de ultrasonidos.	62
Figura 4.2: Rangos de distancias utilizados en la programación.	63
Figura 4.3: Conexionado Motores-L298N-Sensor Shield de Adruino. . .	64
Figura 4.4: Flujograma del programa desarrollado para el movimiento del robot móvil.	66
Figura 4.5: Pines de GPIO de la Raspberry Pi model B.	69
Figura 4.6: Primer método de detección de habitaciones con 2 sensores infrarrojos.	70
Figura 4.7: Esquema conexionado sensor infrarrojo.	71
Figura 4.8: Detección con Haar Cascade Classifier Lower Body.	73

Figura 4.9: Detección utilizando Haar Cascade Classifier Upper Body (superior) y Fallos en la detección (inferiores).	74
Figura 4.10: Fusión de las detecciones a varias escalas para verificar una cara.	75
Figura 4.11: Área de búsqueda de rostros respecto a la imagen completa de cada "frame".	76
Figura 4.12: Funcionamiento del sistema ROS planteado.	78
Figura 4.13: Flujograma del Nodo Talker utilizado en la Raspberry Pi.	79
Figura 4.14: Flujograma del Nodo Listener utilizado en el PC para la monitorización de los datos.	80
Figura 5.1: Distancia de detección con cámara a baja altura y personas de 1,25 metros de altura.	86
Figura 5.2: Distancia de detección con cámara a baja altura y personas de 1,75 metros de altura.	86
Figura 5.3: Distancia de detección con cámara a media altura y personas de 1,25 metros de altura.	87
Figura 5.4: Distancia de detección con cámara a media altura y personas de 1,75 metros de altura.	87
Figura 5.5: Distancia de detección con cámara a altura elevada y personas de 1,25 metros de altura.	88
Figura 5.6: Distancia de detección con cámara a altura elevada y personas de 1,75 metros de altura.	88
Figura 5.7: Momento en que la persona fue detectada por primera vez en habitación con iluminación artificial.	89
Figura 5.8: Momentos en que la persona era detectada en habitación con iluminación artificial.	90
Figura 5.9: Archivo con claras detecciones de cuando se detectaba rostro de la persona al pasar por delante de la cámara.	92
Figura 5.10: Momentos muy desfavorables para la detección de la persona a contra luz.	93

Figura 5.11: Momento detección con persona quieta en la primera posición indicada de habitación pequeña.	95
Figura 5.12: Momento detección de la persona quieta en la segunda posición indicada de habitación pequeña.	95
Figura 5.13: Momento detección con persona quieta en la primera posición indicada de habitación grande.	96
Figura 5.14: Momento detección con persona quieta en la segunda posición indicada de habitación grande.	96
Figura 5.15: Tamaño modelo de rostros (parte superior) y tamaño del área de búsqueda de rostros de cada frame (parte inferior) con el máximo número de rostros teóricamente detectables.	97
Figura 5.16: Turtlebot utilizado para los experimentos.	98
Figura 5.17: Esquema implementación del sistema de detección de personas en un robot Turtlebot.	99

ÍNDICE DE TABLAS

Tabla 4.1: Estado motores según medidas de los sensores. 63

**Tabla 4.2: Lógica de las salidas de Arduino según el estado de los
motores. 64**

Tabla 4.3: Variables para modificar el estado de los motores. 65

ÍNDICE DE FÓRMULAS.

Fórmula 3.1: Fórmula completa para calcular distancia con el sensor. . 51

Fórmula 3.2: Conversión de la velocidad del sonido de m/s a cm/ μ s. . 51

Fórmula 3.3: Fórmula simplificada para calcular la distancia. 51

RESUMEN

Este proyecto trata principalmente de aumentar los campos de la robótica dentro de la monitorización de entornos. Estas aplicaciones, que se intentan conseguir con la robótica, únicamente son métodos de ayudar a las personas quitándoles la carga que supone realizar esos trabajos, por eso se ha decidido realizar este proyecto.

Se pretende crear un robot móvil autónomo que sea capaz de monitorizar la detección de personas en cada una de las habitaciones por las que vaya pasando, de esta forma se tendrá una estimación del número de personas que se detecte en cada una, el tiempo que se tarda en detectar a todas las personas dentro de la habitación y tiempo en recorrer cada habitación.

Este proyecto ayudará para tomar decisiones a las personas o robots como, por ejemplo, cómo reestructurar a sus empleados en las distintas salas, como planificador de tareas a realizar por el robot en función de la ocupación de cada habitación en un horario determinado o simplemente como vigilancia en su propia casa o establecimiento.

El robot consta de un microcontrolador Arduino encargado de realizar el recorrido autónomo por las diferentes estancias del recinto siguiendo la pared que tenga a su izquierda mediante el uso de unos sensores de ultrasonidos para saber la distancia que tiene respecto de las paredes para evitar choques. Cada vez que entre en una habitación se notificará al robot mediante la utilización de un sensor de infrarrojos situados en la parte inferior del robot que se encarga de leer un "código de barras" negro para saber en la habitación a la que está accediendo el robot.

A partir del momento en el que se accede a la habitación, el robot, empezará a detectar rostros de personas utilizando una cámara que está conectada a la Raspberry Pi, que utiliza como ordenador, para lanzar el programa de detección de personas y de la identificación de la habitación en la que se encuentra.

Además el robot se encuentra siempre en comunicación con un ordenador central al que le irán llegando las detecciones de personas y los datos de los sensores infrarrojos de las habitaciones para, a partir de esos valores, estimar la ocupación de cada habitación así como el tiempo que le ha llevado en recorrer cada habitación.

ABSTRACT

This project is mainly to increase the fields of robotics in monitoring environments. These applications, that try to be created using robotics, are only methods of helping people by taking away the burden for doing those jobs, so it's done this Project.

It aims to create an autonomous mobile robot capable of monitoring the detection of persons in each room while it's going past, thus we'll have an estimate of the number of persons detected in them, time takes to detect all persons inside each room and time takes to tour the room.

This project will help to make decisions for people or robots, for example, how to restructure their employees in different rooms, such as scheduler to be performed by the robot based on the occupancy of each room at a specific time or simply as surveillance at home or establishment.

The robot consists of an Arduino microcontroller responsible for carrying out the autonomous tour of the different rooms of the enclosure, following the wall that is on its left by using ultrasonic sensors to know the distance it has to walls to avoid collisions. Once it enters into a room the robot will be notified using an infrared sensor located at the bottom of the robot, which is responsible for reading a black "barcode" to know in the room that is accessing the robot.

From the time when It accesses to the room, the robot will begin to detect human faces using a camera that is connected to the Raspberry Pi, which it's used as a computer to launch the program to detect people and to know the room that is.

Furthermore the robot is always in communication with a central computer, which will receive detections of number of people and the data of infrared sensor from that room, based on these values, to estimate the occupancy of each room as well as the time it has taken to tour each room.

1.INTRODUCCIÓN

En este capítulo se pretende dar a conocer las causas por las que se ha llevado a cabo la necesidad de realizar este robot móvil capaz de monitorizar las detecciones de personas por habitación para conseguir extraer más información del entorno y, de esta forma, utilizar dicha información para dar solución o mejorar las planificaciones de los robots actuales y además se explicaran los objetivos que se buscan cumplir con la realización de este proyecto.

1.1 Motivación.

En los últimos años ha habido un gran avance de la tecnología robótica dentro de la automatización. Cada vez son más las empresas que apuestan por desarrollar robots móviles automatizados para realizar tareas que realizan las personas de forma repetitiva, tareas que suponen un riesgo físico para las personas o cualquier otra función que beneficie o haga más sencilla la vida de la gente.

Un campo muy necesario para el diseño de robots autónomos es el estudio del entorno para la toma de decisiones por parte del robot.

Debido a esta demanda, en el estudio de la semántica del entorno para que el robot realice una navegación más eficiente, se ha desarrollado este proyecto en el que se espera aportar más información como puede ser en este caso la ocupación de una habitación.

La ocupación de las habitaciones puede ser tratada por el robot de múltiples formas para la navegación semántica, la cual consiste en dar más información de elementos del entorno a sistemas de navegación geográficos o topológicos, mejorando de esta forma la navegación al incluir una capa superior de abstracción a estos sistemas. Por ejemplo, se podría utilizar esta información para ser capaz de reconocer estadísticamente en el tipo de habitación que se encuentra dependiendo de la cantidad de gente que detecte en ella, como salas de reuniones, salas de ocio, pasillos, etc, o también para saber el horario en el que puede acceder el robot a las habitaciones por ser más eficiente el encontrarse a menos personas en una habitación para realizar su tarea correspondiente.

1.2 Objetivo.

El objetivo que se quiere cumplir con la realización de este proyecto es la monitorización de la ocupación de personas que existe en las habitaciones.

La ocupación de las personas por habitación puede ser interesante de conocer para el robot, como se ha comentado anteriormente, para tomar decisiones a partir de esa información.

Para ello se utilizará un robot móvil automatizado, que se encargará de recorrer todas las habitaciones, sabiendo en qué habitación se encuentra y el número de detecciones de personas que ha detectado en cada una de ellas, además de saber el tiempo que ha tardado en recorrer cada una de ellas y el tiempo que ha tardado en detectar todas las personas.

Como no se dispone de ningún robot móvil que fuera capaz de recorrer las habitaciones se decidió crear uno.

El robot recorrerá todas las habitaciones utilizando el microcontrolador Arduino y se encargará de controlar el robot para que vaya siguiendo la pared que tenga siempre a su izquierda, de esta forma recorrerá toda la habitación. Para conocer la habitación en la que se encuentra se diseñó un sistema para reconocer la habitación a la que se accede.

Por otro lado, la detección de personas se llevará a cabo utilizando una Raspberry Pi que tomará las mediciones con una cámara utilizando las librerías OpenCV, que se encargará de detectar los rostros de las personas con las que se cruza el robot por las habitaciones.

2. ESTADO DEL ARTE

A continuación se va a hacer un repaso de cómo han ido evolucionando los métodos utilizados para el reconocimiento facial en las últimas décadas hasta llegar a la técnica que actualmente es muy utilizada en diferentes aplicaciones y la cual se utilizará para este proyecto.

Además se realizará un repaso de los proyectos sobre robots móviles que se han visto utilizar últimamente para realizar tareas de monitorización, así como proyectos de monitorización en los que se utiliza principalmente el sistema operativo de robótica ROS.

2.1 Detección de rostros.

Una de las primeras personas que quiso encontrar una explicación en la búsqueda de características de los rostros con las que determinar un teorema para utilizar en los demás casos fue Francis Galton (1822-1911), primo de Charles Darwin.

Galton consiguió determinar la «Ley de Frecuencia del Error» [1] (Teorema del Límite Central). Realizó diferentes experimentos de medición estadística de las características físicas y anatómicas de una gran variedad de individuos, convencido de que en la especie humana existen ciertas características biológicas favorables que son las «más ideales» y que sólo son reunidas en la estructura biológica de un reducido número de individuos, mientras que la gran mayoría de la población se aleja por defecto o por exceso respecto de ese prototipo humano.

La parte superior de la imagen (Figura 2.1) se corresponde con la “media ideal del prototipo de rostro de un delincuente” y la inferior con la “media ideal de rostro de buen hombre”.



Figura 2.1: Imágenes de la media ideal de delincuente y de hombre honrado.

2.1.1 Técnicas.

- El comienzo de las investigaciones en técnicas de reconocimiento de rostros comenzó en los años 60. **Woodrow W. Bledsoe y su equipo** [2] desarrollaron los primeros sistemas semi automáticos de reconocimiento basados en marcado de puntos característicos.
- En los años 70 **Goldstein, Harmon y Lesk** [3] usaron 21 marcadores subjetivos específicos tales como el color del cabello y grosor de labios para automatizar el reconocimiento facial. El problema con estas soluciones previas era que se computaban manualmente.
- En 1988 **Kirby y Sirobich** [4] aplicaron análisis de componentes principales (PCA) [5], al problema del reconocimiento facial. Esto fue considerado un hito al

mostrar que eran requeridos menos de 100 valores para cifrar acertadamente la imagen de una cara convenientemente alineada y normalizada.

- En 1989, **T. Kohoen** [6] definió la teoría de reconocimiento de rostros basada en la caracterización de la cara por la extracción de los autovectores de la matriz de autocorrelación, conocidos como eigenfaces. Daba buenos resultados en caras alineadas pero no para imágenes donde no eran conocidas la situación de la cara.

- En 1991 **Turk y Pentland** utilizando las técnicas Eigenfaces [7], demostraron que el error residual de codificar las eigenfaces se podía utilizar para detectar caras en imágenes no controladas y determinar su posición en las mismas.

Este método combinado con los anteriores resulta de un sistema de reconocimiento fiable capaz de trabajar con imágenes en entornos poco controlados.

- Otras técnicas utilizadas para la detección de rostros es utilizando unos modelos geométricos de la cara [8]. Debido a que todos los rostros siguen unos patrones geométricos con los que se puede decir si en la imagen se encuentra un rostro o no cumpliendo con esos modelos.

- **Scale-invariant feature transform** (o SIFT):

Este algoritmo fue publicado por primera vez por Lowe [9] en 1999, cuyo objetivo es extraer características relevantes de las imágenes para su posterior análisis.

El reconocimiento de rostros [10] ha sido una de las principales aplicaciones de este método gracias a la extracción de características.

Este algoritmo es invariable tanto para cambios de escala como a rotaciones.

- **Algoritmo Viola-Jones (HAAR Cascade Classifier) [11]:**

Es el método que más se está utilizando actualmente, no solo para detectar rostros sino para cualquier tipo de objeto [12] debido a que el clasificador HAAR que se utiliza para la detección de objetos se puede entrenar utilizando imágenes positivas (caras u objeto deseado detectar) y negativas (imágenes sin caras u objetos) para hacer un archivo, llamado clasificador en cascada, con las características personalizadas para el tipo de objeto que se desee localizar en las imágenes.

Estas características son de 3 tipos (Figura 2.2): Borde, Lineal y de Forma o Línea Diagonal.

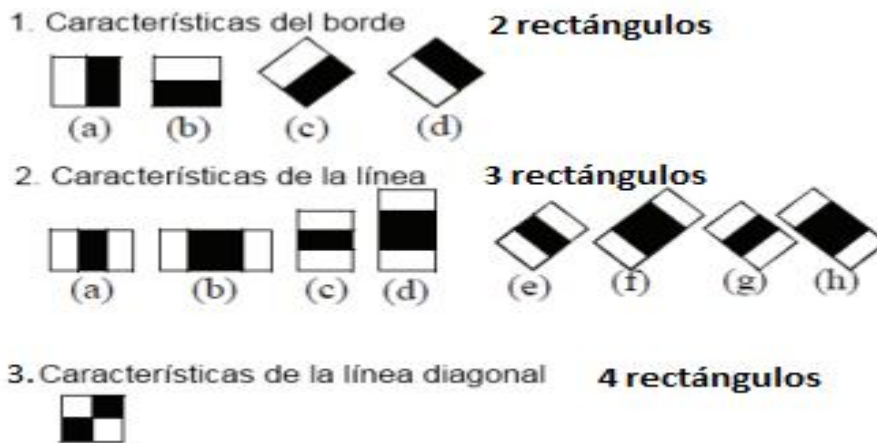


Figura 2.2: Características usadas en clasificadores HAAR.

A continuación se puede observar cómo trabaja el HAAR Cascade Classifier de detección de rostros (Figura 2.3), el cual busca las características contenidas en el clasificador en cascada, una a una, para decidir si es un rostro o no lo es. Para realizar esta comprobación, selecciona una subventana de la imagen del tamaño del clasificador cascada la cual se va moviendo por toda la imagen calculando la característica de tipo Haar de esa subregión para comparar con el umbral dado en la característica Haar de cada nodo de la cascada.

Al ser un Clasificador Cascada, se tienen que cumplir cada una de las características indicadas, sino será descartado el posible rostro.

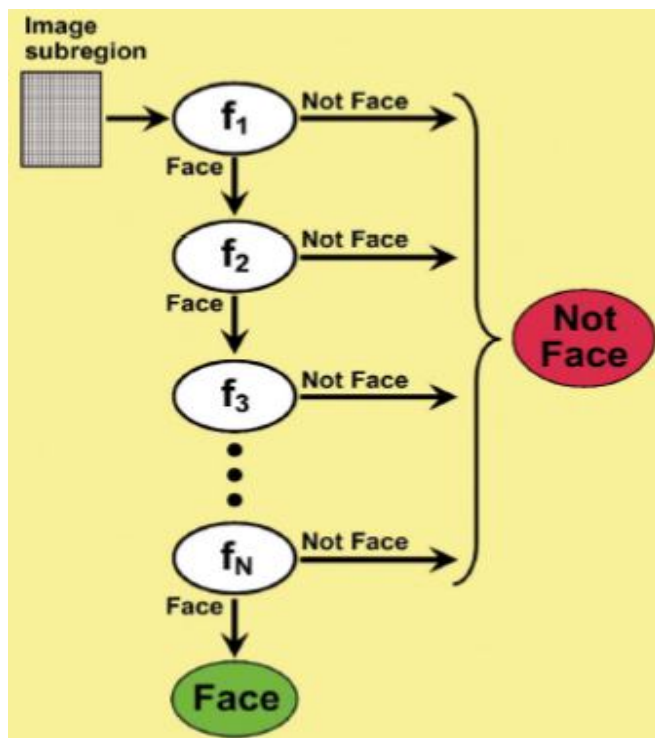


Figura 2.3: Funcionamiento HAAR Cascade Classifier para considerar rostro o no.

La ventaja de este método es la velocidad de cálculo, ya que la detección de las regiones de interés se realizan utilizando las características Haar considerando regiones rectangulares en una ventana de detección, en la que suma las intensidades de los píxeles en cada región y calcula la diferencia entre esas sumas, la diferencia es usada para clasificar subsecciones de la imagen, haciendo posible la detección de rostros en tiempo real.

- **Histograms of Oriented Gradients:**

Este método fue utilizado para detectar personas (de cuerpo entero) por **Navneet Dalal and Bill Triggs** en 2005 [13], pero esta técnica puede ser utilizada para sacar características como puede ser los gradientes de intensidad o dirección de los bordes para identificar cualquier tipo de objeto.

Con este método se consiguen detectar rostros de manera que no influye la pose del rostro ni la iluminación, [14], [15], siendo más efectivo a cambios en el fondo de la imagen.

Existen también técnicas utilizadas para sacar características principales de cada rostro que sirven para realizar un reconocimiento facial para la identificación de cada persona como Principal Component Analysis (PCA) [6], Linear Discriminant Analysis (LDA) [16], Frequency Domain Analysis (FDA) [17], Elastic Bunch Graph Matching (EBGM) [18], entre otros.

Pero en este tema no entraremos en detalle, ya que en este proyecto no nos centramos en identificar a cada persona sino detectar rostros dentro de cada habitación.

2.1.2 Aplicaciones

Con el avance de la detección de personas de los últimos años fueron saliendo aplicaciones importantes como se pueden destacar desde el ámbito de la seguridad ciudadana hasta para reducir el tiempo de espera en los controles de acceso de aeropuertos, aduanas, etc.

A continuación se muestran los campos en los que se está utilizando más la detección de personas y algunas de las soluciones que se han implementado para resolver cada uno de ellos.

- **CONTROL DE ACCESOS.**

El reconocimiento facial y detección de rostros también ha sido muy utilizado para el control de accesos por ser más rápido, disminuyendo las colas de espera, y evitar problemas como olvido de llave, acreditación o contraseña, siendo a día de hoy un método muy demandado.

Una de estas aplicaciones del control de acceso por reconocimiento facial es el control de asistencia a centros educativos mediante terminales de reconocimiento facial (FaceID) [19].

Tiene como finalidad controlar el acceso a un centro educativo, como el del Instituto de Badalona [20], para controlar la asistencia de los alumnos al instituto, pero también posible para el profesorado y el personal del centro.

El reconocimiento facial de esta aplicación se realiza en menos de 2 segundos, la terminal lee, en 3D, el cuadrado formado desde las cejas hasta debajo del labio.

Con esta información, los vectores de distancia y profundidad son suficientes para identificar a la persona.

Mediante la identificación biométrica facial el centro escolar puede llevar un control de los alumnos y profesores que por alguna razón faltan a clase o no llegan a su hora. Además es posible enviar un e-mail o SMS periódicamente, semanal o mensualmente con dichas incidencias.

- **APLICACIONES MÓVILES Y REDES SOCIALES.**

Cada vez están surgiendo más aplicaciones en las que se utiliza el reconocimiento facial es el protagonista por ser un método rápido de identificación del que se puede sacar mucha información.

El reconocimiento facial no solo es muy utilizado hoy en día como se ha visto en el punto anterior para el control de accesos, sino que también se encuentran aplicaciones dedicadas al ocio, como puede ser la aplicación SceneTap [21], startup creada en 2010 por Cole Harper y Marc Doering en Chicago.

Esta aplicación principalmente se encarga de realizar una detección facial utilizando cámaras instaladas en bares y pubs en 13 ciudades de EEUU (Chicago, St. Louis, San Francisco, Austin, Boston, Atlanta, Bloomington, Columbia, Gainesville, Madison, Milwaukee, Phoenix, Athens), por el momento, con las que se pretende identificar los rostros de las personas del establecimiento y sacar información del lugar como: número de personas dentro del local, segmentación por sexos y por edades (Figura 2.4).

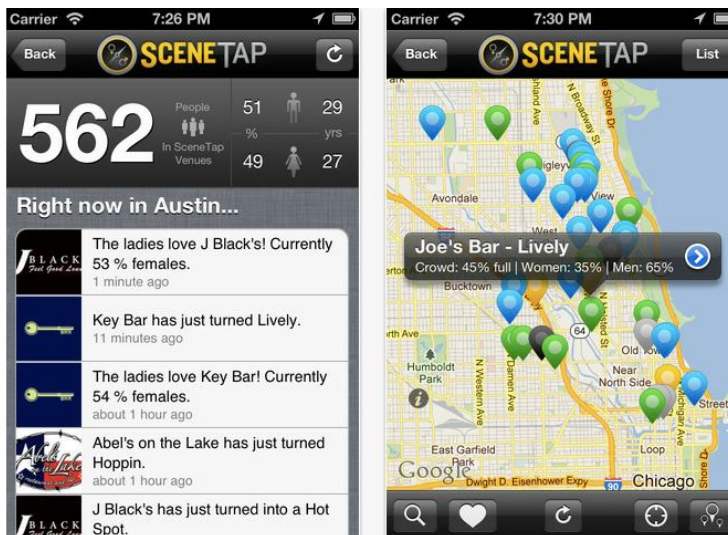


Figura 2.4: Visualización de la app SceneTap.

De esta forma, desde tu aplicación de móvil, puedes saber antes de ir a un establecimiento si este está lleno o no, si hay más hombres o mujeres o si es un local en el que hay más jóvenes o adultos.

Además ellos llaman, en su página web, a la tecnología que utilizan Anonymous Video Analytics (AVA), un sistema de análisis anónimo que cumple las leyes de privacidad [22].

- SISTEMAS DE SEGURIDAD.

El sistema FaceCheck creado por la empresa Cross Match Technologies [23] analiza la entrada de numerosas cámaras de CCTV (Circuito Cerrado de Televisión), y compara las imágenes capturadas con las plantillas y perfiles almacenados en la base de datos.

Este sistema se encarga de detectar y capturar automáticamente la cara de una persona en una base de datos especial. El sistema alerta al operador cuando se detecta un rostro dentro del cuadro de la imagen.

La cámara con reconocimiento facial incorporado situada en la entrada de cualquier negocio puede grabar a todos los que pasan por la puerta principal y conservar sus rostros en la base de datos. Si se detecta un rostro y no se almacena en la base de datos, el sistema desplegará la información necesaria sobre el evento, anunciando instantáneamente la presencia de individuos sospechosos.

La búsqueda de rostro representa un gran ahorro de tiempo en los procedimientos de investigación y en la detección de comportamientos sospechosos.

FaceCheck se ha desarrollado para evitar posibles amenazas y riesgos en diversos establecimientos y asimismo es de gran ayuda en las investigaciones policiales.

Este sistema está diseñado para ser utilizado en diversos escenarios, como se muestra en la descripción [23], como controles fronterizos y edificios que requieren mayor seguridad como pueden ser los casinos [24] para identificar personas “non gratas” o adictas al juego.

- **SEGURIDAD EN LA CONDUCCIÓN.**

La mayor parte de los accidentes de tráfico son debidos al propio conductor (el factor humano) sea de una manera o de otra (falta de pericia, imprudencia, exceso de velocidad, conducir con facultades mermadas, etcétera), así que no es de extrañar que cada vez más fabricantes de automóviles piensen que monitorizar el estado del conductor puede mejorar la seguridad durante la conducción.

En este caso es el grupo PSA Peugeot Citroën [25] el que se suma a la lista y en colaboración con el laboratorio 5 del Centro de Transporte y Procesamiento de Señales de la Escuela Politécnica Federal de Lausanne (EPFL), en Suiza, han desarrollado una tecnología que emplea una cámara de vídeo y un software de reconocimiento facial para captar las diferentes expresiones faciales, parpadeo, movimientos musculares, y emociones del rostro del conductor.

El sistema está permanentemente vigilante y en un primer momento pensaron en hacer un sistema que reconociera si el conductor estaba distraído, si estaba sufriendo somnolencia o si no estaba en condiciones de seguir conduciendo.

Saber todas estas expresiones supone un problema ya que cada persona puede actuar de formas distintas, por lo que se optó por el momento, para simplificar la tarea de identificación, realizar un seguimiento de sólo dos expresiones: las vinculadas a la ira y el disgusto, cuyas manifestaciones son similares.

- **BIOMETRÍA FORENSE.**

Los softwares de reconocimiento facial, actualmente son un método de identificación de sospechosos muy demandado.

La empresa NEC ha desarrollado unos sistemas de reconocimiento facial a partir de imágenes digitales (NeoFace Reveal [26]) o por cámaras de seguridad en tiempo real (NeoFace Watch [27]).

El software de reconocimiento facial, diseñado para identificar automáticamente sospechosos comparando cualquier imagen digital obtenidas con las fotos alojadas en su base de datos, se ha utilizado oficialmente por el cuerpo de policía de Chicago, en 2013 [28], desde que se consiguió culpar al delincuente Pierre Martin, identificado con el software con una foto del perfil del sospechoso realizada con una cámara de seguridad del tren en el que cometió un robo con pistola en Enero de 2013 y en Febrero de 2013 en el mismo lugar y, además, es utilizado por agentes de policía de

Leicestershire a partir de mediados de 2014, primer equipo del Reino Unido en probar el software de reconocimiento facial NeoFace.

Este nuevo software reduce la cantidad de tiempo que los agentes tienen que pasar consultando su base de dato, ya que es muy rápido, al buscar coincidencias por regiones del rostro, lo cual funciona para imágenes con baja resolución y con fotografías de perfil.

2.2 Monitorización de entornos.

En este apartado se va a hacer un repaso de los robots móviles y sistemas robóticos actualmente más conocidos utilizados para la monitorización del entorno.

Es importante distinguir que los sistemas robóticos que se van a estudiar se corresponden a sistemas fijos, estáticos, que realizan unas tareas colocando los sensores y actuadores en una posición concreta, mientras que lo referido a robots móviles, utilizados para monitorizar entornos, son robots cuyo movimiento por el entorno se realiza de forma manual o autónoma e incorpora una serie de sensores para realizar toda la monitorización.

2.2.1 Robots móviles monitorizados.

Los robots móviles monitorizados son robots que realizan unas tareas concretas, en concreto, dichos robots se utilizan principalmente para realizar tareas que no pueden realizar humanos, por ser peligrosas o imposibles para nosotros, o tareas que son realizadas más rápido y más precisas por robots que si las hicieran los humanos.

Entre los robots móviles monitorizados más destacados en la actualidad se encuentran los rovers de exploración en Marte como el Mars Science Laboratory (MSL)[29], más conocido como Curiosity, robots militares utilizados en catástrofes o robots domésticos.

- **Mars Science Laboratory o Curiosity:**

El Curiosity es el proyecto lanzado por la NASA en 26 de Noviembre de 2011 desde Cabo Cañaveral, Florida, llegando a Marte el 6 de Agosto de 2012 [30], cuyos cuatro objetivos principales a cumplir en Marte, como se indican en la web dedicada a los objetivos a cumplir y ya cumplidos por el rover [29], dichos objetivos son imposibles actualmente de realizarlos los humanos en otro planeta por esta razón se envían este tipo de rovers para cumplirlos:

Determinar si existió vida alguna vez en Marte, caracterizar el clima de Marte, determinar su geología y prepararse para la exploración humana de Marte, de los que ya se han respondido a algunas de estas preguntas en los experimentos que se han llevado a cabo. [31, 32].

Para contribuir a estos cuatro objetivos científicos y conocer el objetivo principal (establecer la habitabilidad de Marte) el Curiosity tiene ocho cometidos.

- Evaluación de los procesos biológicos:

1.º Determinar la naturaleza y clasificación de los componentes orgánicos del carbono.

2.º Hacer un inventario de los principales componentes que permiten la vida: carbono, hidrógeno, nitrógeno, oxígeno, fósforo y azufre.

3.º Identificar las características que representan los efectos de los procesos biológicos.

- Objetivos geológicos y geoquímicos:

4.º Investigar la composición química, isotópica y mineral de la superficie marciana.

5.º Interpretar el proceso de formación y erosión de las rocas y del suelo.

- Evaluación de los procesos planetarios:

6.º Evaluar la escala de tiempo de los procesos de evolución atmosféricos.

7.º Determinar el estado presente, los ciclos y distribución del agua y del dióxido de carbono.

- Evaluación de la radiación en superficie:

8.º Caracterizar el espectro de radiación de la superficie, incluyendo radiación cósmica, erupciones solares y neutrones secundarios.

Para lograrlos se dispuso al rover de los siguientes instrumentos [33] para llevar a cabo la monitorización de todas las tareas a cumplir:

- Cámaras:

MastCam: Este sistema proporciona imágenes en múltiples espectros y en color real a través de cámaras con visión 3D.

Mars Hand Lens Imager (MAHLI): Este sistema consiste en una cámara montada en un brazo robótico del rover, y se usará para obtener tomas microscópicas de las rocas y suelo marciano.

MSL Mars Descent Imager (MARDI): Durante el descenso a la superficie marciana MARDI tomó imágenes en una distancia de 3.7 kilómetros hasta los 5 metros de altura respecto del suelo. Permite hacer un mapeo del terreno circundante y del sitio de aterrizaje.

Hazard Avoidance Cameras (Hazcams): Utiliza cuatro pares de cámaras en blanco y negro, para realizar la navegación, situadas en la parte delantera, izquierda, derecha y trasera del vehículo. Estas imágenes de salvaguarda sirven para que el vehículo no choque contra obstáculos inesperados.

Navigation Cameras (Navcams): Utiliza dos pares de cámaras de navegación en blanco y negro 3-D montadas sobre el mástil de apoyo para el desplazamiento sobre el terreno.

- Espectrómetros:

ChemCam: ChemCam es un sistema de espectroscopia inducida por rayo láser, el cual puede apuntar a una roca a una distancia de 13 metros, vaporizando una pequeña cantidad de los minerales subyacentes en ella y recogiendo con la cámara el espectro de luz emitida por la roca vaporizada.

Espectrómetro de rayos X por radiación alfa (APXS): Este dispositivo irradiará muestras con partículas alfa y permitirá su análisis a partir del espectro generado por los rayos X reemitidos.

CheMin: Instrumento de análisis químico y mineralógico a través de la difracción y fluorescencia de rayos X, el cual cuantifica y analiza la estructura de los minerales contenidos en una muestra.

Análisis de muestras en Marte (SAM): El instrumento así denominado, analizará muestras sólidas y gaseosas en búsqueda de compuestos orgánicos. El espectrómetro láser es capaz de medir ratios de isótopos de carbono y oxígeno en el dióxido de carbono.

- Detectores de radiación:

Detector por evaluación de radiación (RAD): Este instrumento analizará toda la gama e intensidad de radiación espacial y radiación solar que recibe la superficie de Marte.

Albedo dinámico de neutrones (DAN): DAN es una fuente pulsante de neutrones, la cual será utilizada para medir la concentración de hidrógeno o agua bajo la superficie.

- Sensores medioambientales:

Estación de supervisión ambiental rover (REMS): Esta es una estación meteorológica que medirá la presión atmosférica, humedad, dirección y fuerza del viento, así como la temperatura ambiental y los niveles de radiación ultravioleta.

- **Robots para ayuda en catástrofes y militar:**

Actualmente la ayuda de los robots para misiones, que suponen un alto riesgo para la salud de los humanos, es cada vez más demandada y se conocen casos como tras el accidente nuclear de Fukushima, en el que por la alta radiación producida por las explosiones y daños en los reactores era peligroso para los humanos acceder al complejo. Los robots como iRobot 510 PackBot, Quince, ASTACO-SoRa, entre otros, comenzaron con el reconocimiento de peligros y limpieza de la zona [34].

A continuación, se va a hablar de uno de estos robots utilizados para esa misión.

- iRobot 510 PackBot [35]:

Es un robot diseñado por la empresa iRobot para la realización de misiones peligrosas tanto militares como de seguridad ciudadana (Figura 2.5), es un robot modular, ampliable y robusto.



Figura 2.5: Robot 510 PackBot.

Entre sus principales misiones, cabe destacar:

- Desactivación de bombas / desactivación de artefactos explosivos.
- Detección e identificación de materiales peligrosos Químicos, Biológicos, Radioactivos y Nucleares (CBRN).
- Tareas de vigilancia y de reconocimiento.

Para la realización de estas misiones y las demás de las que es capaz de realizar este robot, posee los siguientes sensores [36] para detectar los diferentes peligros:

- **Smiths Detection LCD 3.3:** Sensor capaz de detectar Agentes de Guerra química y productos químicos tóxicos industriales.
- **RAE Systems MultiRAE Pro:** Sensor para detectar multi-amenaza.
- **FLIR Fido XT:** Sensor para detectar el rastro de explosivos.
- **Canberra UltraRadiac Plus:** Utilizado para la detección de radiación.
- **Thermo Scientific FirstDefender RMX:** Detecta agentes químicos desconocidos.
- **Drager X-am 5600:** Sensor para la detección de gases químicos.
- **Thermo Scientific RadEye GN+:** Sensor para la detección aumento del nivel de radiación gamma y de neutrones.
- **FLIR Systems nanoRaider ZH:** Utilizado para detectar e identificar niveles de radiación gamma y de neutrones.

Además posee una serie de cámaras para la detección y manipulación de materiales peligrosos:

- Cámara principal: Destinada para la inspección, proporciona 312x zoom y tiene una combinación de luces blancas y de infrarrojos para la iluminación en condiciones de poca luz.
- Cámara Térmica: Situada junto a la cámara principal para proporcionar visión térmica en las inspecciones.

Estos son algunos de los dispositivos más importantes de los que dispone el robot, aunque dispone de muchos más [37] que no se han comentado para la realización de las misiones e imprescindibles para las funciones del robot.

• Robots Domésticos:

Un claro ejemplo de la aparición de los robots en la realización de tareas cotidianas en la vida de las personas son los robots aspiradores autónomos (Figura 2.6). Los cuales se encargan de la limpieza del suelo de las diferentes estancias de la casa, sabiendo en todo momento en la habitación que está, cuánto han limpiado y lo que les queda por limpiar.



Figura 2.6: Robot aspirador.

Los primeros robots que aparecieron y que actualmente se siguen utilizando en robots de "gamas bajas" se mueven por unos patrones de movimiento prefijados, los más comunes suelen ser espiral, zig-zag, forma de S, poligonal, siguiendo las paredes, etc. Aunque los más avanzados de hoy en día utilizan un sistema de navegación inteligente, que les permite crear un mapa de la habitación, capaz de saber qué zonas ha limpiado y cuáles no, evitando movimientos incoherentes y consumos innecesarios en la batería.

Los sistemas más utilizados para realizar el mapeo, patrones de limpieza y localización en estos tipos de robots son:

- **Sistema de doble cámara:** Utiliza una cámara superior para explorar con mayor precisión y realiza un mapa de los suelos y techos incluso con las luces apagadas y su cámara inferior lee el suelo para medir la distancia de desplazamiento y reconoce los pequeños cambios que puedan existir, este sistema lo utilizan robots como el LG VR6270LVM [38].
- **Sistema IAdapt:** Un software que utilizan los robots Roomba [39] que, apoyado en sus sensores, determina 40 conductas distintas y es capaz de tomar hasta 60 decisiones por segundo. Este tipo de sistemas, más complejos, dotan al robot de una conducta basada en la adaptación al espacio y al grado de suciedad.
- **Sistemas Mapa Láser:** Basado en un mapa generado por láser. El láser inspecciona todos los objetos que hay dentro de una habitación y crea un mapa de la misma, una vez que el robot está ubicado dentro del mapa, aspira siguiendo un patrón predeterminado, este sistema mide el espacio pero no la suciedad. Este sistema es conocido en los robots Neato que utilizan el sistema BotVision [40], cuyo sistema utiliza un sensor láser de 360° colocado en la parte superior del robot.

2.2.2 Sistemas robóticos de monitorización.

Los sistemas robóticos de monitorización, que se hacen referencia en este apartado, son sistemas estáticos que requieren estar situados en determinados sitios y en una posición concreta para tomar medidas del entorno y realizar alguna función.

A continuación se va a hablar de uno de los sistemas robóticos más utilizados en la monitorización que más influencia tiene con este proyecto: los sistemas domóticos, inmóticos y urbóticos.

Esta es una de las aplicaciones más demandadas hoy en día para el confort y seguridad de los edificios como viviendas, edificios residenciales y de trabajo, entre otros.

Los principales objetivos que se quieren cumplir con la automatización de viviendas y edificios son el control de la luminosidad, de la calefacción/aire acondicionado y de gestión de la seguridad mediante alarmas técnicas.

Actualmente existen proyectos y documentación para hacer casas o habitaciones domóticas [41] utilizando diferentes estándares de programación con los que se puede automatizar diferentes procesos como pueden ser el control de la luminosidad con el encendido y apagado de la luz dependiendo de la luminosidad de la habitación, control de calefacción y aire acondicionado en función de la temperatura captada por termostatos, control sobre dispositivos multimedia como música y TV, video vigilancia con la utilización de cámaras web o cámaras IP y además, todo ello, puede ser visualizado y controlado desde dispositivos externos con acceso a internet.

Pero el principal problema es que, para una misma aplicación como puede ser el control de la luminosidad de la casa, vigilancia, o cualquier otra tarea de monitorización, se requiere tener los mismos sensores en todas las estancias por ser un sistema estático.

Con la realización de este proyecto se quiere solucionar estos problemas que conlleva utilizar estos sistemas robóticos estáticos en los que, por ejemplo, supone un mayor coste económico robotizar todas las estancias en las que se desee monitorizar el entorno para realizar la misma función en todas las estancias que, en cambio, con un robot móvil, provisto de todos los elementos necesarios para la monitorización de las variables del entorno que se quieran estudiar, podría recorrer periódicamente las estancias y realizar la monitorización en cada una de ellas.

2.3 Empleo de Raspberry Pi y Arduino en la robótica.

Originalmente Raspberry Pi y Arduino fueron proyectos pensados principalmente como herramientas de enseñanza, en las que cualquier persona pueda tener acceso a ellas al ser herramientas de muy bajo coste.

Por un lado, Raspberry Pi (Figura 2.7) es un ordenador de placa reducida o (placa única) (SBC) de bajo coste, desarrollado en Reino Unido por la Fundación Raspberry Pi lanzada a la venta el 29 de Febrero de 2012, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

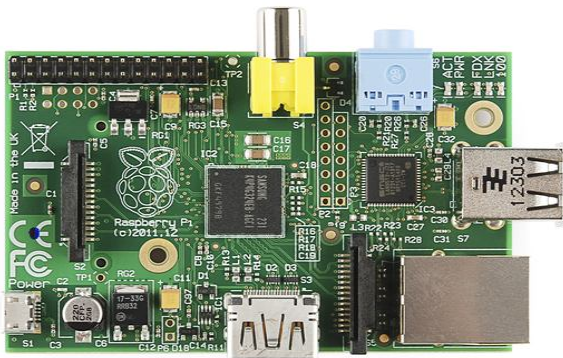


Figura 2.7: Raspberry Pi.

Gracias al hardware que posee la Raspberry Pi [42], a pesar de su bajo coste, posee un procesador de hasta 700MHz y RAM de 512MB que hace posible el uso de un Sistema Operativo, mayoritariamente Sistemas Operativos basados en el núcleo Linux y, por tanto, se considera un ordenador.

Además de tener un procesador y una memoria RAM suficientes para poder usar un SO, la Raspberry Pi posee una GPIO (Figura 2.8) (General Purpose Input/Output), pines de entrada/salida, con la que podemos configurar para interactuar con el exterior de la Raspberry Pi mediante sensores, motores, actuadores o cualquier dispositivo que pueda ser controlado mediante salidas digitales.



Figura 2.8: Pines GPIO de la Raspberry Pi.

Por otro lado, Arduino (Figura 2.9) fue desarrollado como una plataforma de hardware libre lanzada en 2005 [43], basado en un microcontrolador y un entorno de desarrollo.

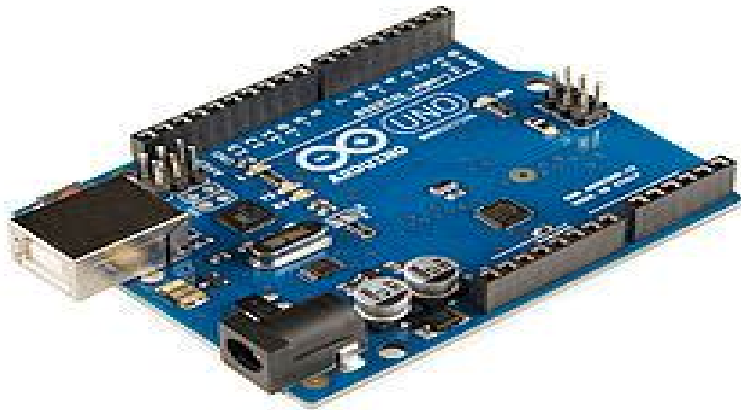


Figura 2.9: Arduino modelo UNO.

El microcontrolador usado puede ser desde 8 bits hasta el más novedoso de ellos de 32 bits.

El software consiste en un entorno de desarrollo, el cual se comunica con el microcontrolador utilizando el protocolo STK500 [44] que utiliza cabeceras en lenguaje C.

Arduino puede tomar información del entorno interactuando con el exterior a través de sus entradas analógicas y digitales, controlando luces, motores y otros actuadores. Además una de las ventajas de utilizar un microcontrolador como Arduino es que los proyectos realizados pueden ejecutarse sin necesidad de conectar a un computador.

2.3.1 Aplicaciones monitorizadas.

Las posibilidades a la hora de realizar proyectos con Raspberry Pi y Arduino son infinitas como hemos visto anteriormente además de ser asequible económicamente y fácil de usar.

En el caso de Arduino son muy conocidos los proyectos de robótica y monitorización, entre otros como se verá a continuación, al tratarse de un microcontrolador y su gran cantidad de entradas/salidas analógicas y digitales.

Si a todas estas posibilidades que nos aportan Arduino, al ser un microcontrolador barato con bastantes entradas y salidas tanto digitales como analógicas, lo juntamos con Raspberry Pi, al tratarse de un ordenador, para realizar un proyecto, las opciones para realizar distintos proyectos son infinitas.

A continuación se muestran algunas de las aplicaciones existentes para la monitorización de entornos.

- **RASPBERRY PI.**

Existen proyectos en los que se utiliza la Raspberry Pi como un Web Server en el que se monitorizan las detecciones, realizadas por la Raspberry Pi, desde la web para ser interpretadas remotamente desde otro PC o cualquier dispositivo con acceso a internet.

Un ejemplo de ello es el proyecto de **Priyanga y Ramanan** [45], en el que se implementa un sistema de video de búsqueda de supervivientes para implementar en un vehículo aéreo no tripulado. Dicho sistema (Figura 2.10) se encarga de monitorizar desde la web las medidas de voltaje del nivel de batería, posición del vehículo aéreo con el GPS y transmisión del video capturado por la cámara.

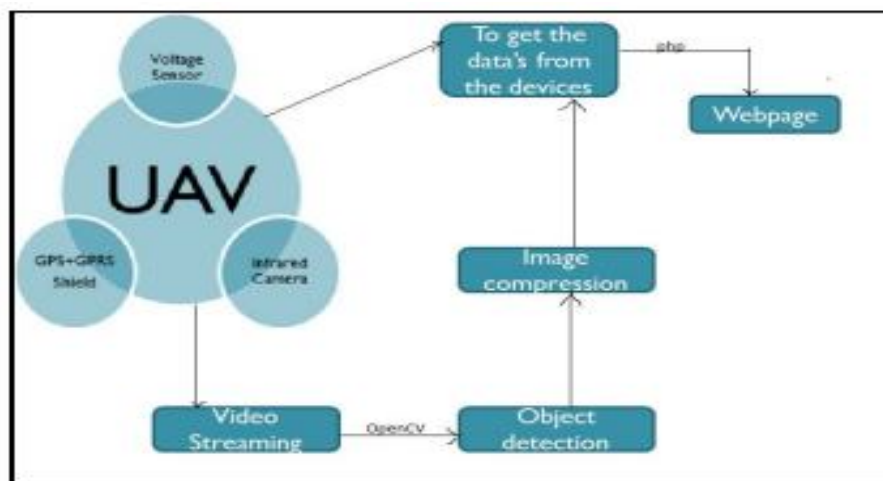


Figura 2.10: Sistema propuesto por Priyanga y Ramanan.

- **ARDUINO.**

Existen incontables aplicaciones de todo tipo de monitorizaciones utilizando Arduino de las que podemos destacar algunos de los temas más utilizados con este microcontrolador como puede ser la utilización del microcontrolador para aplicaciones domóticas [46], para saber cómo se encuentra el entorno en una habitación y actuar en consecuencia, monitorización eléctrica, para saber el consumo eléctrico [47], o monitorización de las condiciones ambientales [48].

Un ejemplo de ellos es el proyecto, realizado por **Patrick Di Justo y Emily Gertz**, para monitorizar el medioambiente [48].

Este proyecto consiste en medir diferentes variables climatológicas utilizando sensores conectados a las entradas de Arduino. Se pueden detectar gases como: carbon monoxide, carbon dioxide, liquid petroleum gas, butane, propane, methane (natural gas), hydrogen, ethyl alcohol, benzene, volatile organic compounds, ammonia, ozone y hydrogen sulfide.

Para la detección del vapor de agua utiliza sensores infrarrojos. En la detección de aerosoles se utilizan un receptor de luz verde y rojo que dependiendo de la cantidad de voltaje producido por el LED Rojo y LED Verde estará el cielo despejado (si producen el mismo voltaje) o habrá neblina o ambiente con aerosol (se detecta menos verde por lo que produce menos voltaje en LED Verde).

- **RASPBERRY PI y ARDUINO.**

Existen proyectos en los que se unen Raspberry Pi y Arduino para realizar sistemas domóticos como el que se encuentra en el libro de Dennis [49], el cual describe como realizar la monitorización de la luz y de la temperatura de la estancia en donde se alojen los sensores para controlar una serie de elementos.

De esta forma, se utilizan los pines del Arduino para conectar los sensores y tener más posibilidad de albergar un número mayor de ellos y de actuadores, ya que la Raspberry Pi no dispone de tantos pines I/O.

En este proyecto se utiliza un Termistor como termómetro para saber la temperatura de la estancia. Sabiendo la temperatura se controla un ventilador con un Relé encendiendo o apagando dicho ventilador si se sobrepasa una temperatura determinada.

Además se utiliza una Fotorresistencia para detectar la luminosidad de la habitación, y de esta forma se abre o se cierra con un motor DC la cortina de la ventana en caso de detectar poca o mucha luz.

2.3.2 Robots móviles.

Uno de los campos más abarcados dentro de los proyectos ya existente de Raspberry Pi y Arduino es el de la robótica.

A continuación se describirán diferentes proyectos de robots móviles, proyectos individuales de Arduino o con Raspberry Pi y también un proyecto utilizando ambas plataformas.

- **ARDUINO.**

A continuación se muestra un proyecto de robot móvil propuesto en el libro de **Margolis** [50].

En este proyecto se explica cómo se puede diseñar un robot móvil con motores DC, el cual se mueve con las técnicas más utilizadas en este tipo de robots con Arduino, mediante sensores IR y de ultrasonidos.

Estos robots móviles (Figura 2.11) están programados para ser independientes y no tener que controlarlos manualmente. Se utilizan los sensores infrarrojos para moverse siguiendo una línea negra representada en el suelo y además posee los sensores de ultrasonidos para evitar obstáculos midiendo la distancia al obstáculo.

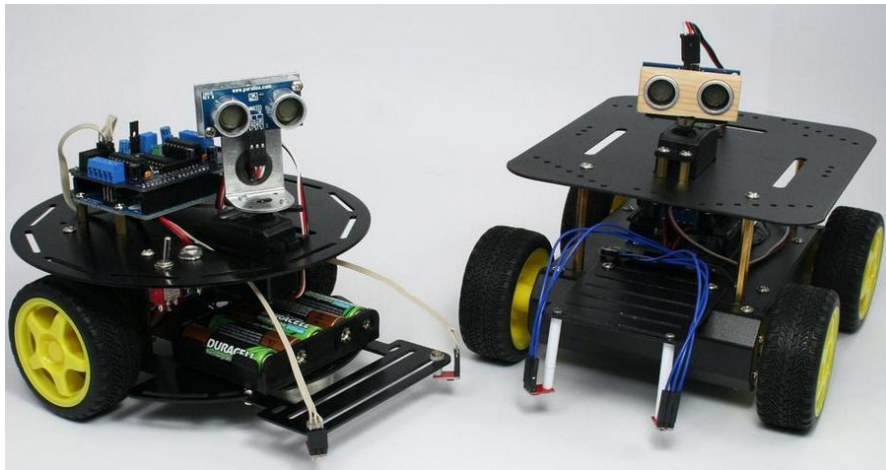


Figura 2.11: Robots móviles desarrollados en el libro de Margolis.

El funcionamiento de estos robots se basa en dotar al robot de dos sensores infrarrojos con un emisor y un receptor cada uno de ellos, los cuales, detectan la línea oscura al no recibir la luz infrarroja reflejada en ella y de esta forma se sabe si te estas separando de la línea o no.

En caso de separarse de la línea, se detectará por los sensores y se realizarán las maniobras programadas correspondientes.

Los motores son controlados por los pines de salida del Arduino que están conectados a un puente H que es el que se encarga de alimentar los motores en la etapa de potencia.

- **RASPBERRY PI.**

En este caso se ha seleccionado un libro de Richard Grimmett [51] para mostrar un ejemplo de un proyecto de un robot móvil (Figura 2.12) que describe el desarrollo para realizar robots controlados a través de una Raspberry Pi.

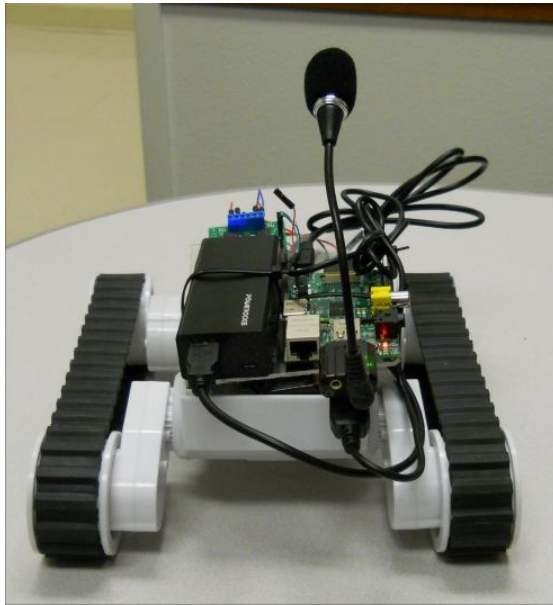


Figura 2.12: Robot Móvil propuesto en el libro de Grimmett.

Dentro de los proyectos que propone este libro, se incluyen, entre otros, crear un robot móvil con motorización DC de las ruedas, dotar al robot de micrófono para el control del mismo por voz, cámara para detección de color, sensores infrarrojos para detectar obstáculos, etc.

- **RASPBERRY PI Y ARDUINO.**

Utilizar el microcontrolador Arduino y Raspberry Pi para realizar un proyecto de robótica es una buena solución, ya que contrarrestan sus desventajas con las ventajas del otro al favorecer la rapidez del procesador y SO de Raspberry Pi y además Arduino puede controlar más pines de salida y más pines de entrada por lo que podemos dotar a nuestro robot de más sensores y actuadores.

Un ejemplo de ello es el siguiente proyecto de Jesús Vico [52], el cual se plantea crear un robot controlado de 2 modos distintos:

- Control Manual:

Se controla remotamente utilizando los sensores de aceleración de eje X, Y y Z de un smartphone Android que enviará dichas medidas por “topics” de ROS a la Raspberry Pi que controla el robot móvil y esta manda las instrucciones al Arduino por comunicación Puerto Serie para que el microcontrolador actúe en consecuencia sobre los motores que controla según las órdenes dadas por la Raspberry Pi.

- Control Autónomo:

El control del robot móvil se realiza evitando los obstáculos que tuviera por delante.

Para realizar esto se dota al robot de 5 sensores de ultrasonidos HC-SR04 para cubrir 180° de la visión delantera del coche. Este método consiste en enviar la distancia de los sensores al nodo Master (Raspberry Pi) mediante “topics” de cada uno de los sensores y será la Raspberry Pi quien dé las órdenes de movimiento de los motores al Arduino.

2.4 ROS.

El Sistema Operativo Robótico (ROS) [53] es uno de las principales herramientas usadas en este proyecto como se verá más adelante.

ROS proporciona bibliotecas específicas para los robots, además de servicios como abstracción del hardware, control de dispositivos de bajo nivel, etc.

También ofrece la comunicación entre procesos mediante la utilización de mensajes (Publisher y Subscriber) y la configuración de parámetros.

Esta comunicación se puede utilizar en diferentes contextos como pueden ser los siguientes:

- **Monitorización en entornos virtuales para misiones de rescate.**

Aunque el libro de Gonçalo Nuno dos Santos [54] está pensado para monitorizar casos que se dan para monitorizar un entorno virtual de simulación, este sistema también puede ser llevado a la realidad.

En este proyecto se toman sobre un entorno virtual de simulación medidas de visualización del entorno para reconocer el entorno y paredes.

Además de medidas de radiofrecuencia y de la voz para detectar personas en situaciones de peligro para una rápida detección.

- **Monitorización de los topics.**

No solo se utilizan topics en ROS para monitorizar el entorno, sino que el proyecto de **Jeff Huang y su equipo** de la Universidad de Illinois [55] pretenden monitorizar todos los topics desde un framework para garantizar la seguridad de esos topics.

Esto quiere decir, que si los topics no cumplen unas políticas o no concuerdan con las especificaciones del robot, se podrán modificar o eliminar para que no haya problemas y de esta forma evitar que se boicotee el robot o cualquier ataque informático sobre el robot al tratarse ROS de un sistema opensource distribuido.

- **Planificación del movimiento.**

Otro proyecto realizado con ROS es el de **Kalakrishnan y su equipo** [56] el cual se ha creado un algoritmo para predecir la trayectoria del movimiento que menos coste computacional tenga para el robot para evitar los obstáculos que se encuentre. Para ello realiza una serie de simulaciones con todas las posibilidades para luego realizar la que le convenga más.

- **Teleoperación de robots.**

En el proyecto de **Matei Ciocarlie, Kaijen Hsiao, Adam Leeper y David Gossow** [57] **Mobile Manipulation Through An Assistive Home Robot** se desarrolla un robot móvil con el que se puede teleoperar para manipular los objetos de la casa utilizando librerías de ROS para la ayuda a personas con movilidad reducida.

Cuenta con una cámara 3D para localizar objetos y mediante una interfaz (Figura 2.13), el usuario, puede seleccionar con un click el objeto que se quiere coger.

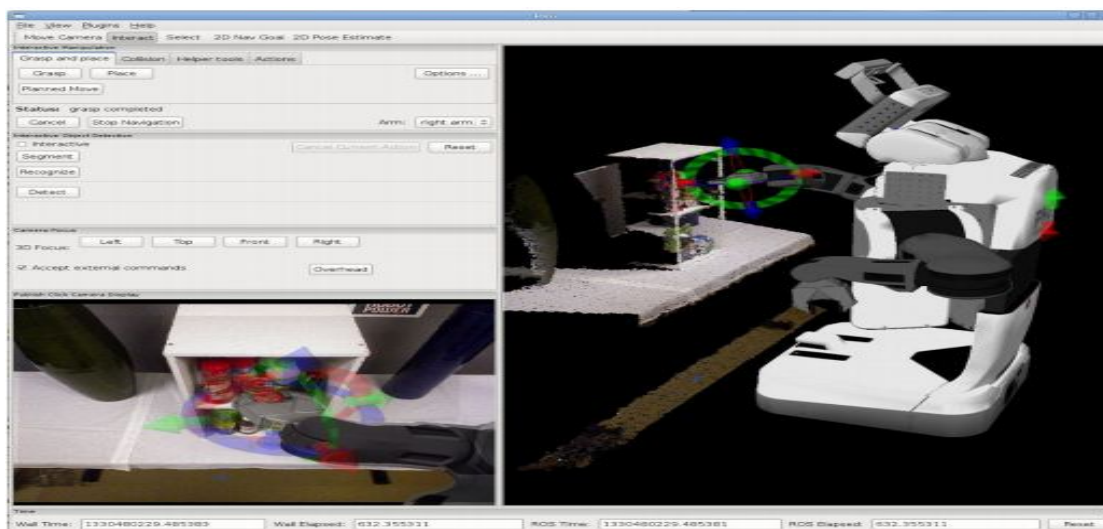


Figura 2.13: Aplicación monitorización y teleoperación con robot.

- **Detección de objetos transparentes.**

En el proyecto de **U. Klank, D. Carton y M. Beetz** [58] se presenta el problema de detectar objetos transparentes como vasos de cristal y botellas de plástico, en los que los sensores de infrarrojos tienen problemas de detectar estos objetos al atravesar la luz los objetos.

A este problema dan una solución utilizando ROS y una cámara Kinect que toma imágenes en 3D desde muchas posiciones para luego reconstruir los objetos (Figura 2.14) que hay en la imagen que son transparentes.



Figura 2.14: En las imágenes se pueden ver los resultados de las detecciones.

- **Monitorización y comunicación en un sistema domótico.**

ROS como se ha visto es utilizado para múltiples aplicaciones de las que además se puede utilizar para proyectos de domótica como el proyecto de **Philip Parsch** [59]. Se establece la comunicación mediante topics entre los nodos para enviar mensajes para el control de los dispositivos y monitorización para realizar la visualización de los estados de los dispositivos (Figura 2.15).



Figura 2.15: Aplicación de visualización y monitorización de los dispositivos.

3. SISTEMA DESARROLLADO

El sistema que se ha desarrollado para la realización del robot móvil debe cumplir el siguiente diagrama de casos de uso (Figura 3.1). Un usuario puede interactuar con el sistema de monitorización y/o robot móvil autónomo de la siguiente forma: un usuario puede iniciar la monitorización del entorno, iniciar el robot para que realice un recorrido autónomo por las estancias y puede leer por pantalla los datos de las monitorizaciones por habitación.

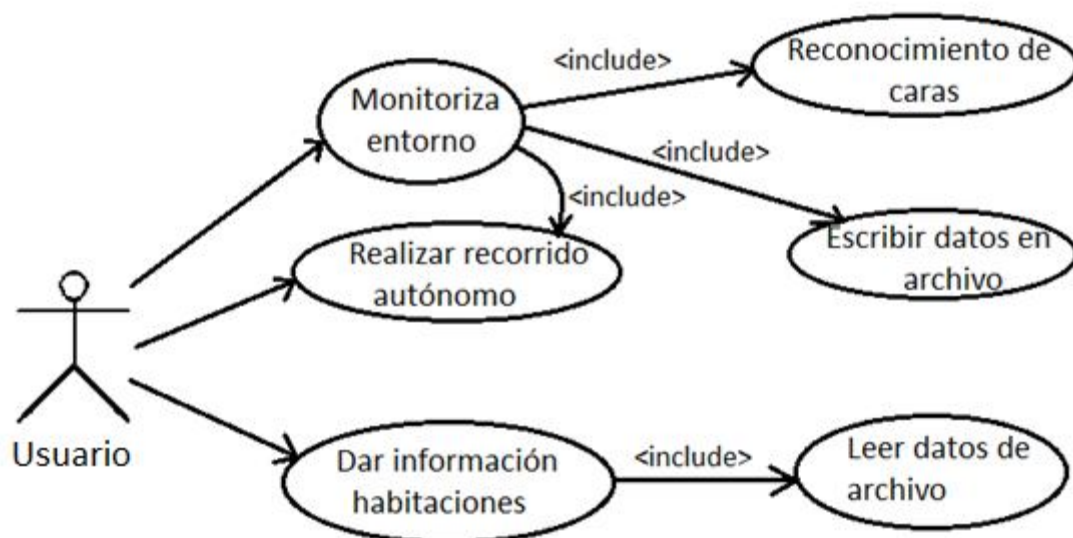


Figura 3.1: Diagrama de casos de uso entre el usuario y el sistema desarrollado.

Además, para cumplir las especificaciones del sistema dadas por el diagrama de casos de uso, la programación de todo ello se puede ver de forma resumida en el siguiente diagrama de clases (Figura 3.2), en el cual se aprecian las clases utilizadas para realizar la programación y las funciones principales utilizadas.

Se tiene un “Sistema” propuesto en el que se realiza toda la monitorización de habitaciones del que forman parte de él dos clases independientes, “Monitorización de habitación” y un “Sistema Navegación”. En la “Monitorización de habitación” se realizan todas las monitorizaciones que se hacen en las habitaciones, en este caso “Reconocimiento Facial”. Por otro lado, en el “Sistema de Navegación”, se tienen dos elementos para realizar la navegación por habitaciones, uno es la utilización de un “Sensor Habitación” que en este caso se utilizará un sensor “Infrarrojo” y el otro es una clase para realizar un “Seguimiento de contorno” de la habitación en el que para realizar este seguimiento requiere de la utilización de un “Sensor de Distancia” y el control de unos “Motores”.

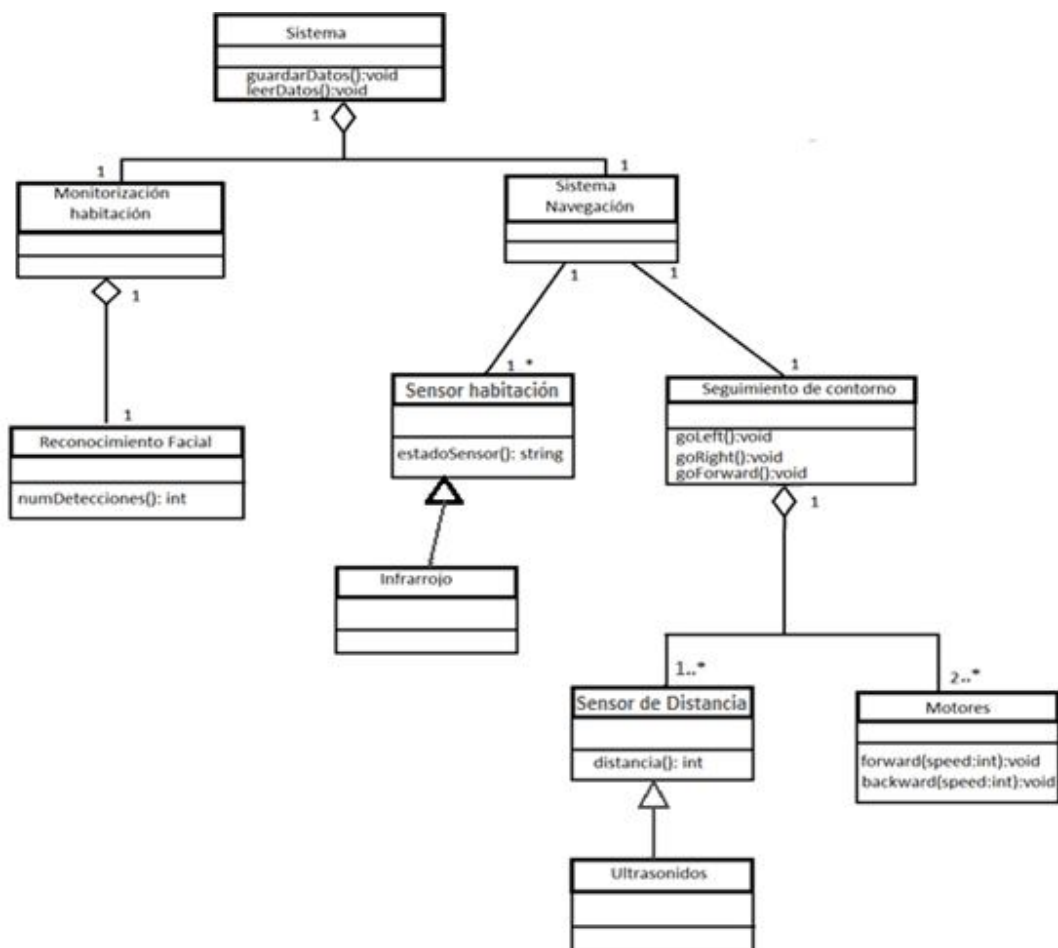


Figura 3.2: Diagrama de clases del sistema propuesto.

Por otro lado, desde el punto de vista de hardware, para realizar el robot móvil autónomo capaz de monitorizar se ha decidido utilizar el microcontrolador Arduino UNO [60] para el manejo autónomo del coche que se encarga de realizar el recorrido de todas las habitaciones siguiendo la pared que tenga a su izquierda, utilizando dos

sensores de ultrasonidos. Además el coche dispone de una Raspberry Pi modelo B, sin conectarla al Arduino, la cual, se encarga de realizar las detecciones de rostros para conocer el número de personas que hay en cada habitación y además toma medidas con un sensor de infrarrojos que debe leer un código de barras a la entrada de cada habitación para saber por qué habitación está pasando.

Dichas detecciones (personas detectadas y estado del sensor de infrarrojos) son enviadas mediante topics de ROS a una estación de trabajo, en este caso un ordenador portátil, que será donde se encuentre la lógica del proyecto para quitarle coste computacional a la Raspberry Pi.

Este programa que recopila las detecciones de la Raspberry Pi en el PC debe reconocer la habitación en la que se encuentra el robot según los estados del sensor infrarrojo y también determinará el número máximo de personas detectadas en cada una de las habitaciones, así como el tiempo que ha tardado en saber cuántas personas hay en la habitación y el tiempo transcurrido desde que entra hasta que sale de cada habitación.

Para comprender todo esto de mejor forma, a continuación se explicará detalladamente el diseño y elementos elegidos para el robot (Figura 3.3).



Figura 3.3: Robot móvil creado con sistema de detección de rostros.

3.1 Parte 1: Robot móvil.

En esta parte se va a explicar los elementos que se han utilizado y la descripción de ellos a la hora de crear el robot móvil autónomo que va a recorrer las habitaciones siguiendo el recorrido de la pared situada siempre a su izquierda.

La parte del robot móvil tiene unas dimensiones de 21,5x14x11,5 cm y está compuesto de plástico para reducir su peso. El peso total del conjunto del robot es de 1,230 Kg. El robot realiza el trayecto por las habitaciones mediante la utilización de 2 sensores de ultrasonidos para seguir la pared situada a su izquierda, manteniéndose a una distancia entre 5 y 20 cm de la pared, y además de evitar obstáculos que se encuentre de frente. El robot realiza el recorrido por habitaciones a una velocidad de 0.2 metros/segundo en línea recta, recorriendo cualquier tipo de estancia como, por ejemplo, habitaciones de 10 m² en unos 2,5 minutos de media.

3.1.1 Chasis del Robot.

En la siguiente imagen (Figura 3.4) se puede apreciar la estructura del robot.

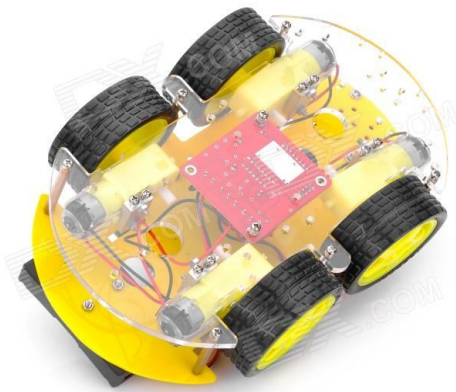


Figura 3.4: Chasis Robot (Vista boca abajo).

Se compone de 2 placas de plástico, una encima de la otra, separadas por 4 cm para dejar espacio entre medias de ellas (Figura 3.5) para colocar el Arduino y el puente H con todo el cableado. De esta forma se ahorra espacio para poder colocar en la parte superior del robot la Raspberry Pi, la cámara, los sensores de ultrasonidos para las paredes, el adaptador Wifi y la batería que alimentará al robot.

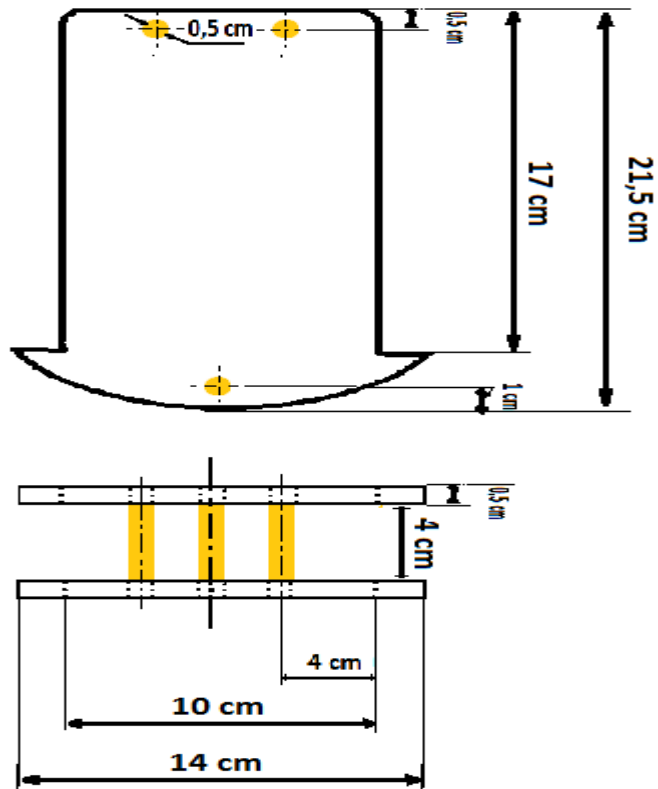


Figura 3.5: Esquema medidas placas del Chasis.

Las ruedas utilizadas (Figura 3.6) son de plástico con goma para disminuir el peso con este material y facilitar el movimiento. A continuación se muestran las ruedas donde se han seleccionado 2 pares de ruedas en cada lado, todas movidas por motores DC, para que el giro del robot sea más solidario y más rápido que si se hiciera utilizando únicamente dos ruedas.



Dimensiones:

Diámetro: 6,5 cm.

Ancho: 2,5 cm.

Figura 3.6: Ruedas del robot.

3.1.2 ETAPA DE POTENCIA (L298N)

Para poder controlar correctamente los motores DC utilizando Arduino, se debe colocar una etapa de potencia entre medias de los pines de salida de Arduino y los motores para poderlos alimentar con la corriente suficiente. Esto es necesario ya que, Arduino no puede proporcionar en sus pines de salida la corriente suficiente para que funcionen los motores porque los pines de salida solo pueden llegar a una corriente de 40 mA como máximo.

Por lo que debemos utilizar una etapa de potencia. En este caso se ha escogido esta placa con L298N (Figura 3.7) ya que es una placa que según las especificaciones cumple correctamente los requisitos para controlar los 4 motores DC.

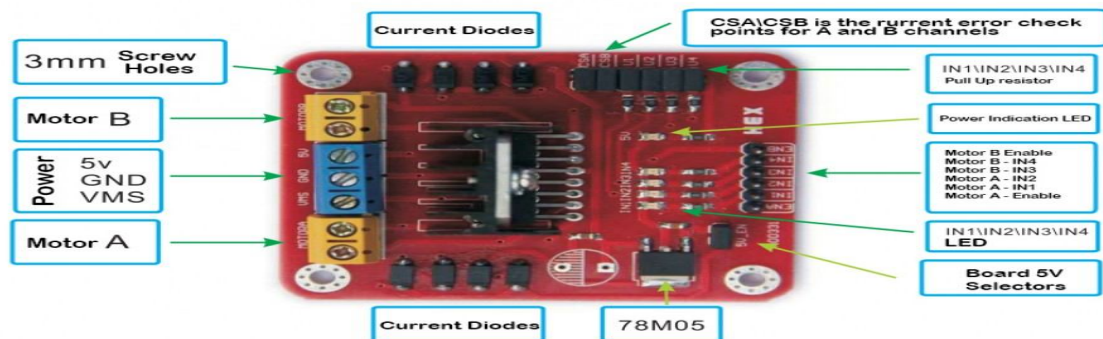


Figura 3.7: Placa puente H L298N.

Especificaciones:

- Corriente máxima salida motores: 2 A.
- Tensión alimentación pines motores: 5V.

El L298N se ha colocado en la placa inferior de plástico del chasis del robot, como se muestra en la imagen siguiente (Figura 3.8), utilizando 4 tornillos para mantenerlo sujeto.

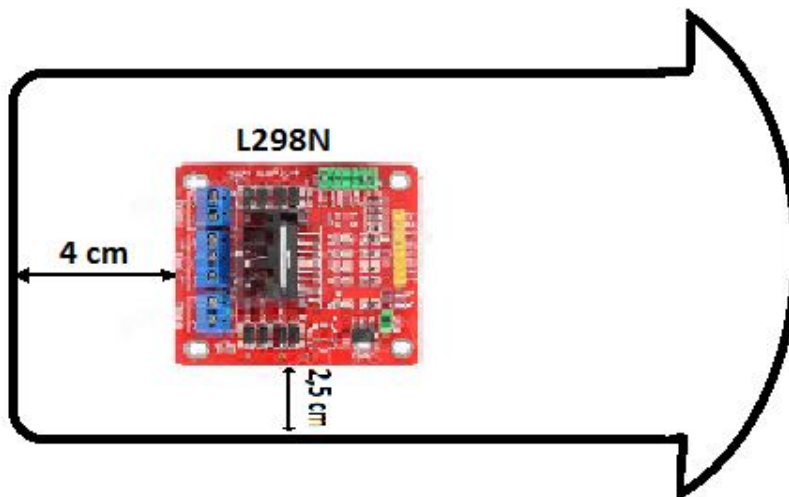


Figura 3.8: Esquema de colocación del L298N en el chasis.

3.1.3 SENSORES ULTRASONIDO (HC-SR04)

La detección de proximidad, para que siga las paredes y mantener el robot pegado a ella, se realiza con dos sensores de ultrasonidos. Serán dos sensores de ultrasonidos porque es el número mínimo de ellos para poder seguir las paredes, evitar colisiones frontales y el más económico.

Se han utilizado sensores ultrasónicos porque son de los sensores más fiables, ya que no intervienen agentes externos como puede ser la iluminación, al contrario que si se hubieran utilizado, por ejemplo, sensores de infrarrojos que dependen de la iluminación.

En este caso se han elegido los sensores HC-SR04 (Figura 3.9) al ser un sensor que cumple con la función que tienen que realizar, ya que detectan medidas entre 2 centímetros y 4,5 metros.



Especificaciones:

Dimensiones: 4,5x2x1,5 cm.

Tensión de trabajo: 5VDC

Corriente de reposo: <2mA

Angulo: <15°

Distancia: 2 a 450cm

Resolución: 0,3cm

Figura 3.9: Sensor ultrasonidos HC-SR04.

El funcionamiento para calcular distancias parte de la utilización de dos transductores (componentes electrónicos capaces de transformar una magnitud física en otra, en este caso de sonido a electricidad y viceversa).

Uno actúa como un altavoz y lanza un pulso de ultrasonidos que, al impactar contra un objeto, produce un eco. El otro transductor actúa como un micrófono y escucha este eco (Figura 3.10).

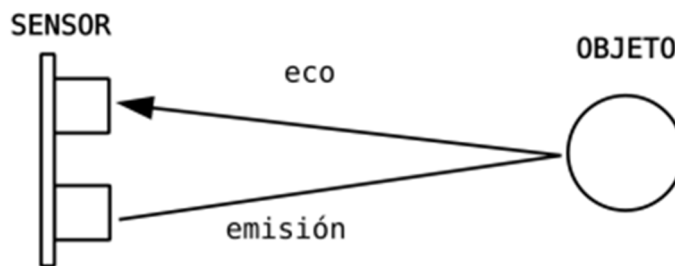


Figura 3.10: Funcionamiento sensor de ultrasonidos.

El funcionamiento de este sensor se logra a partir de 2 pines digitales: el pin de trigger (disparo) y echo (eco). El primero recibe un pulso de habilitación de parte del microcontrolador, mediante el cual se le indica al microcontrolador que comience a realizar la medición de distancia. A través de un segundo pin (echo) el sensor “muestra” al microcontrolador un pulso cuyo ancho es proporcional al tiempo que tarda el sonido en viajar del transductor al obstáculo y luego de vuelta al módulo (Figura 3.11).

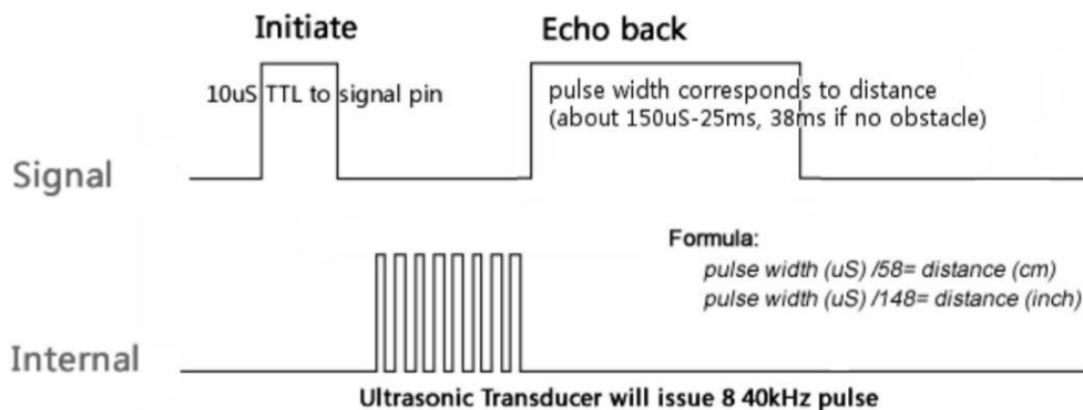


Figura 3.11: Esquema de señales enviadas y recibidas por el sensor.

Mediante una sencilla formula (Fórmula 3.1) puede estimarse entonces la distancia entre el sensor y el obstáculo si se conoce el tiempo de viaje del sonido y el tiempo que ha tardado el micrófono en recibir el eco.

$$\text{Distancia} = V \cdot \frac{t}{2} = \frac{340 \frac{m}{s} \cdot Echo}{2}$$

Fórmula 3.1: Fórmula completa para calcular distancia con el sensor.

Dónde la "V" es la velocidad del sonido en el aire (340 m/s) y "t" el tiempo que tarda en llegar el eco al micrófono. Lo dividimos por dos ya que la señal recorre dos veces la distancia entre el objeto y el sensor.

El tiempo de Echo viene dado en microsegundos y la distancia que se desea medir es en centímetros, por tanto se pasa la velocidad del sonido en el aire a centímetros/microsegundos (Fórmula 3.2).

$$340 \frac{m}{s} * \frac{100 \text{ cm}}{1 \text{ m}} * \frac{1 \text{ s}}{10^{-6} \mu s} = 0,034 \frac{cm}{\mu s}$$

Fórmula 3.2: Conversión de la velocidad del sonido de m/s a cm/μs.

Resumiendo, se quedaría la ecuación (Fórmula 3.3) a utilizar en el programa de Arduino para medir distancias de la siguiente forma:

$$\text{Distancia (cm)} = \frac{\text{Echo } (\mu s) * 0,034 \left(\frac{cm}{\mu s} \right)}{2} = \frac{\text{Echo } (\mu s)}{58 \left(\frac{\mu s}{cm} \right)}$$

Fórmula 3.3: Fórmula simplificada para calcular la distancia.

A continuación se muestra la disposición (Figura 3.12) que se ha estudiado para colocar los sensores de la forma más eficiente.

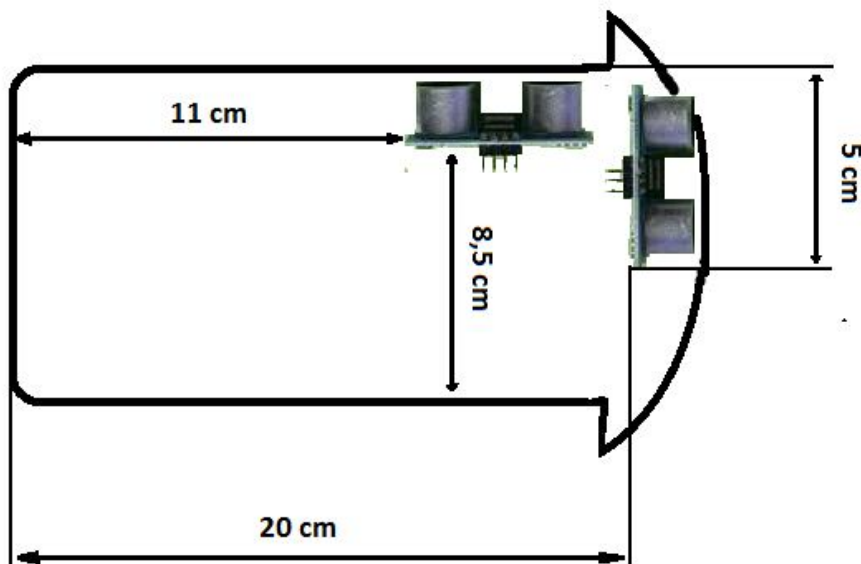


Figura 3.12: Esquema colocación sensores ultrasónicos.

Dicha colocación se ha realizado en la placa superior de plástico, el segundo piso del chasis, para estar en la parte externa del robot y evitar falsos positivos con las ruedas de la parte inferior.

La colocación se ha hecho de esa forma porque así el sensor de la izquierda mantendrá al robot pegado cerca de la pared, irá buscando la pared, y el sensor de delante se utiliza para evitar colisiones frontales. Se colocan en forma de L y juntos para evitar huecos como por ejemplo que haya un hueco a la izquierda y sea demasiado tarde para girar, de esta forma al estar juntos, se podrá girar rápidamente a cualquier cambio en la pared de la izquierda que va siguiendo.

La sujeción de los sensores a la placa de plástico se puede hacer de la forma que mejor convenga, aunque en este caso se ha hecho con unos alambres de plástico para evitar más volumen y peso al robot.

3.1.4 Arduino UNO.

El microcontrolador elegido para el control del robot es Arduino UNO. Se ha elegido por ser suficiente para poder controlar el robot, ya que de momento son necesarias 4 salidas digitales para el control de los motores y otras 4 entradas digitales para los dos sensores de ultrasonidos (1 trigger y 1 echo cada uno) lo que da un total de 8 pines digitales quedando todavía libres 6 pines digitales y 6 analógicos para posibles ampliaciones de sensores del robot futuras, por este motivo no se ha utilizado una placa con más I/O como Arduino Mega, ya que con el modelo UNO es suficiente para este proyecto.

Estas son algunas de las características principales de Arduino UNO:

- Microcontrolador: ATMEGA328
- Voltaje de Operación: 5V
- No. de Entradas/Salidas Digitales: 14
- No. de Salidas PWM: 6
- No. de Entradas Analógicas: 6
- Corriente DC por Entrada ó Salida: 40mA
- Velocidad de Reloj (Clock Speed): 16MHz
- Dimensiones: 7,5 x 5,3 x 1,5 cm.

Como se aprecia en la corriente DC de salida, será necesario la utilización de una etapa de potencia para alimentar a los motores con corriente suficiente.

La colocación del Arduino se realiza en la primera placa de plástico (Primer piso), colocado al lado del puente H L298N en la posición que se indica a continuación (Figura 3.13). Para su sujeción se ha atornillado a la placa de plástico.

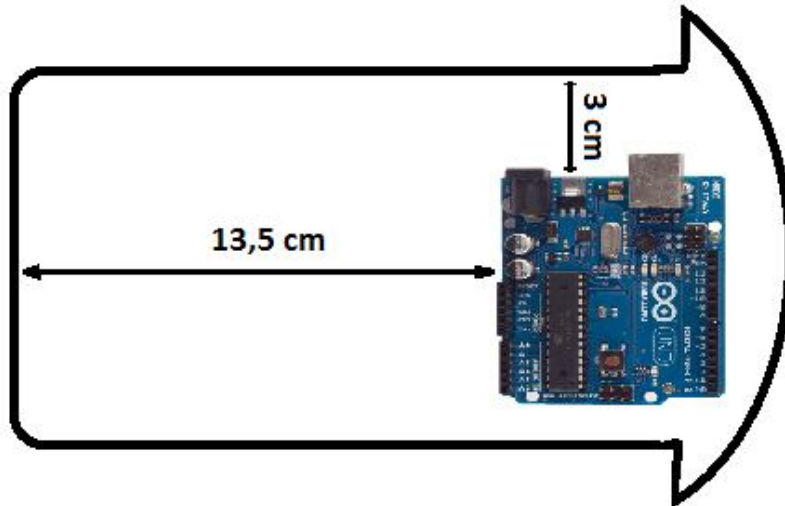


Figura 3.13: Esquema colocación Arduino UNO.

3.1.5 Arduino Sensor Shield.

Además de utilizar un Arduino UNO, se ha optado añadir un Sensor Shield de Arduino (Figura 3.14), ya que los sensores de ultrasonidos que se utilizan para medir la distancia a las paredes utilizan 3 pines cada uno (Voltaje, Tierra y Señal) y por tanto el Arduino Uno dispone únicamente de 3 pines de Tierra, 1 pin de 5 V y 1 pin de 3,3V.

Se podría aumentar el número de pines de Voltaje y de Tierra con una protoboard o de alguna otra forma, pero se pensó que utilizando el Sensor Shield de Arduino nos ahorraríamos crear otra solución y además de esta forma se puede reutilizar para futuros proyectos en los que se quiera aumentar el número de sensores para el robot.

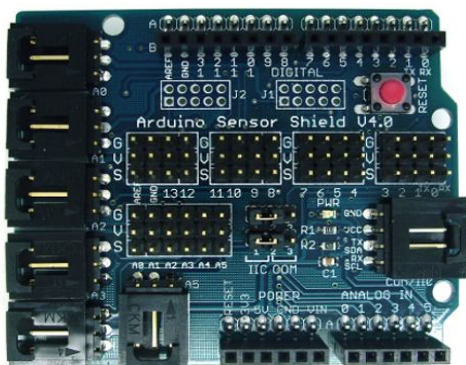


Figura 3.14: Sensor Shield de Arduino.

La colocación del Sensor Shield se realiza colocándolo justo encima del Arduino uniendo ambos por los pines de conexión. A partir de ahora todas las conexiones que se realicen se harán en este Sensor Shield.

3.2 Parte 2: Detección de personas y habitaciones.

Las funciones que cumple la Raspberry Pi en el proyecto son: un sistema automático de detección de rostros, reconocimiento de la habitación en la que se encuentra y comunicación inalámbrica de dichas detecciones con el exterior (PC).

Las características principales que incluye la incorporación de este sistema al robot móvil es el aumento de la altura del robot, ya que se eleva la cámara como se explicará a continuación para detectar rostros y por tanto se pasa de los 11,5 cm a un robot de 31,5 cm de altura. Además con este sistema se consigue unas detecciones a 5 Hz recibiendo todos los datos en una estación de trabajo externa al robot.

Todo este sistema se realiza mediante la utilización de los siguientes elementos.

3.2.1 Raspberry Pi.

Como ya se ha explicado anteriormente, Raspberry Pi es básicamente un ordenador. Con este ordenador dotamos al robot con Sistema Operativo con la capacidad de lanzar nuestros programas para detectar rostros con la cámara conectada por USB, habitación en la que se encuentra el robot y comunicación con ordenador externo.

Se ha utilizado Raspberry Pi porque posee unas características suficientes para la realización de este proyecto, además de ser una opción muy económica. Existen otras alternativas que dan mejores prestaciones, como pueden ser, por ejemplo, Cubieboard o Banana Pi con mejores procesadores y memoria RAM, pero también aumenta su coste.

Como solo se va a trabajar con el uso de la webcam, comunicación por Wifi y 2 pines de entrada de la GPIO para los sensores infrarrojos de color para saber la habitación en la que se encuentra, son más que suficientes las características de una Raspberry Pi, en concreto se utilizará el modelo B que cuenta con algunas de las siguientes prestaciones:

- Procesador: 700 MHz ARM1176JZF-S core CPU.
- GPU: Broadcom VideoCore IV.
- Memoria RAM: 512 MB
- Low-Level Peripherals:
 - 8 x GPIO
 - I2C bus
 - +3.3V
 - +5V
 - Ground
- Power Requirements: 5V @ 700 mA via MicroUSB.
- Dimensiones: 8.6cm x 5.4cm x 1.7cm.

La colocación de la Raspberry Pi se va a llevar a cabo en la parte superior (Figura 3.15), en la segunda placa de plástico (Segundo Piso), con una sujeción con tornillos.

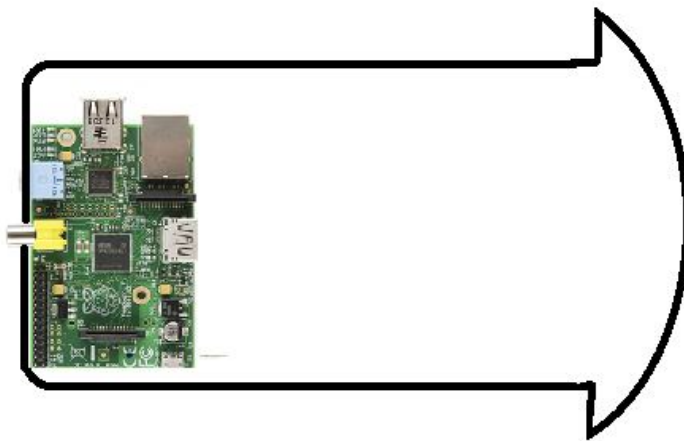


Figura 3.15: Esquema colocación Raspberry Pi en placa superior del Chasis.

3.2.2 Sensor infrarrojo.

Para saber en qué habitación ha entrado el robot se ha pensado en utilizar un método basado en el sistema del código de barras, que se explicará con más detalle más adelante, el cual, requiere detectar el color negro que se encontrará dibujado en el suelo.

Se requiere utilizar para ello un sensor de infrarrojo por ser los sensores más económicos y que dan buenos resultados para la detección del color negro.

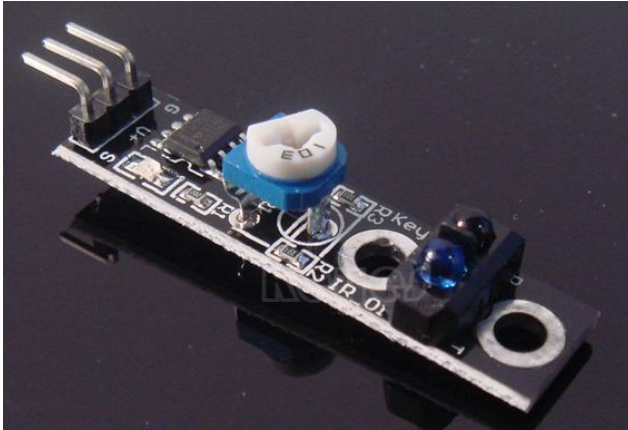


Figura 3.16: Sensor infrarrojos Keyes IR-01.

El modelo utilizado es el Keyes IR-01 (Figura 3.16) que utiliza el sensor TCRT5000 infrarrojo además cuenta con un potenciómetro que se puede usar para ajustar la sensibilidad. A continuación se muestran las siguientes características para detectar el color negro:

Especificaciones:

- Distancia Detección: 0-3 cm.
- Voltaje funcionamiento: 2.5V - 12V. Recomendable usar 5V.
- Corriente funcionamiento: 18 - 20mA.

Dicho sensor se necesita colocar en la parte inferior del robot, junto a las ruedas (Figura 3.17), porque se quieren reconocer los códigos que habrán representados en el suelo cuando el robot pase sobre ellos.

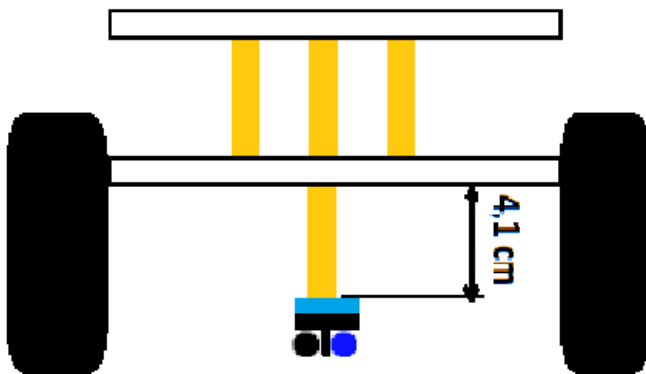


Figura 3.17: Esquema colocación sensor infrarrojo.

3.2.3 Adaptador Wifi

Se va a necesitar utilizar para la comunicación entre Raspberry Pi y ordenador externo un adaptador Wifi para poder mandar las mediciones llevadas a cabo por la cámara y los sensores de infrarrojos.

Para ello se puede usar cualquier adaptador Wifi que sea compatible con Raspberry Pi pero que, en este caso, disponemos del adaptador Belkin F5D7050 (Figura 3.18) conectado por USB.



Figura 3.18: Adaptador Wifi Belkin F5D7050.

Este adaptador Wifi es compatible con redes 802.11g permitiendo una velocidad de transferencia media de 22 Mbps utilizando el ancho de banda de 2,4 GHz.

La configuración del Wifi con el SO utilizado en la Raspberry Pi se explicará detalladamente más adelante.

3.2.4 Cámara

Se necesita una cámara para utilizar con la Raspberry Pi para la detección de rostros que se va a realizar utilizando las librerías OpenCV.

Para ello se va a reutilizar una vieja cámara de la Playstation 2, el EyeToy modelo Namtai (Figura 3.19), la cual, utiliza en la Raspberry Pi el módulo gspca_ov519 tomando frames a 5 Hz para no saturar el procesamiento de imágenes evitando "timeouts" innecesarios.

Se ha elegido esta cámara ya que se disponía de ella y al ser compatible con la Raspberry Pi se ha optado por reutilizarla, pero se podría haber utilizado cualquier otra webcam compatible para realizar el mismo software de detección de rostros que se va a desarrollar.



Figura 3.19: Cámara EyeToy Namtai.

La cámara fue colocada sobre una bobina de DVDs para así conseguir detectar mejor las personas al conseguir una mejor altura. Además la cámara se colocará apuntando en sentido contrario al del sensor de ultrasonidos izquierdo (Figura 3.20).

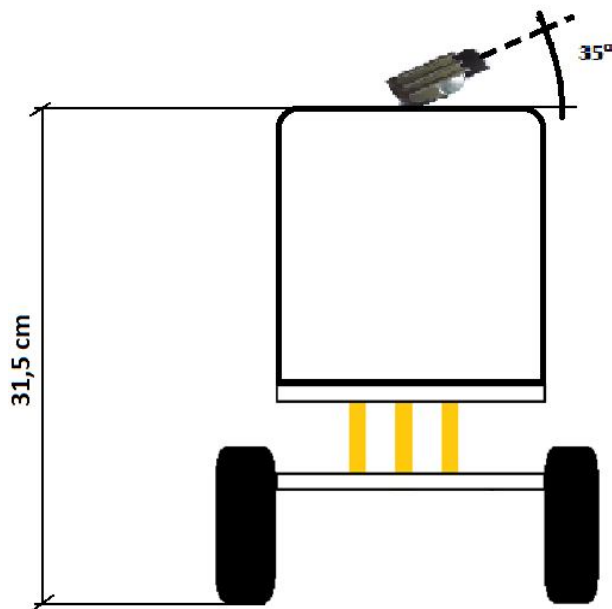


Figura 3.20: Esquema colocación cámara sobre bobina de DVDs en robot.

3.3 Batería.

Para el funcionamiento autónomo e inalámbrico del robot, se ha utilizado una batería que sea capaz de alimentar de forma simultánea al robot móvil y el sistema de detección de rostros y habitaciones.

La batería que se debía elegir tenía que poder alimentar con 5V tanto al robot móvil como al sistema de detección, ya que ambos trabajan con el mismo voltaje de alimentación.

Dentro de las baterías disponibles se decidió adquirir la RAVPower (Figura 3.21), que dispone de 10.400mAh y 2 salidas de 5V (una de 1 A y otra de 2 A) suficientes para alimentar al robot móvil y al sistema de detección de personas y habitaciones respectivamente.



Figura 3.21: Batería RAVPower de 10.400mAh con salidas a 5V de 1A y 2A.

Dispone de unas dimensiones de 11,8 x 7 x 2,1 cm y un peso de 227 g, la cual se situará en la parte superior del robot y bajo la cámara ya que es la posición del robot cuyo peso menos influye al robot, menos obstaculiza los cables y posición de los sensores y además ayuda a elevar un poco más la cámara para tomar mejores detecciones.

4. DESARROLLO DEL SISTEMA.

En este capítulo se va a describir el desarrollo que se ha llevado a cabo para lograr realizar el proyecto, funcionando correctamente tanto la parte del robot móvil como la detección de rostros por habitación.

La descripción del desarrollo del sistema se realiza por separado porque son sistemas independientes en el proceso de desarrollo. Por un lado, robot móvil y, por el otro, el sistema de detección de rostros y habitación junto con la comunicación con la estación de trabajo en la que se centralizarán todas las monitorizaciones.

4.1 Parte 1: Robot móvil.

En este apartado se explica lo realizado con Arduino para el diseño del robot móvil encargado de moverse por las habitaciones.

En la descripción de los elementos utilizados para Arduino (sensores de ultrasonidos y placa de puente H para la etapa de potencia de los motores) se describió la función y colocación de ellos sobre el chasis del robot.

A continuación se va a pasar a explicar el conexinado de todo el sistema con Arduino y la programación de este.

4.1.1 Sensores de Ultrasonido

El conector que se ha elegido realizar entre los sensores de ultrasonido es el mostrado en la imagen (Figura 4.1). Se han conectado al Sensor Shield de Arduino, como se explicó en el capítulo anterior, por disponer de más pines de 5V y GND.

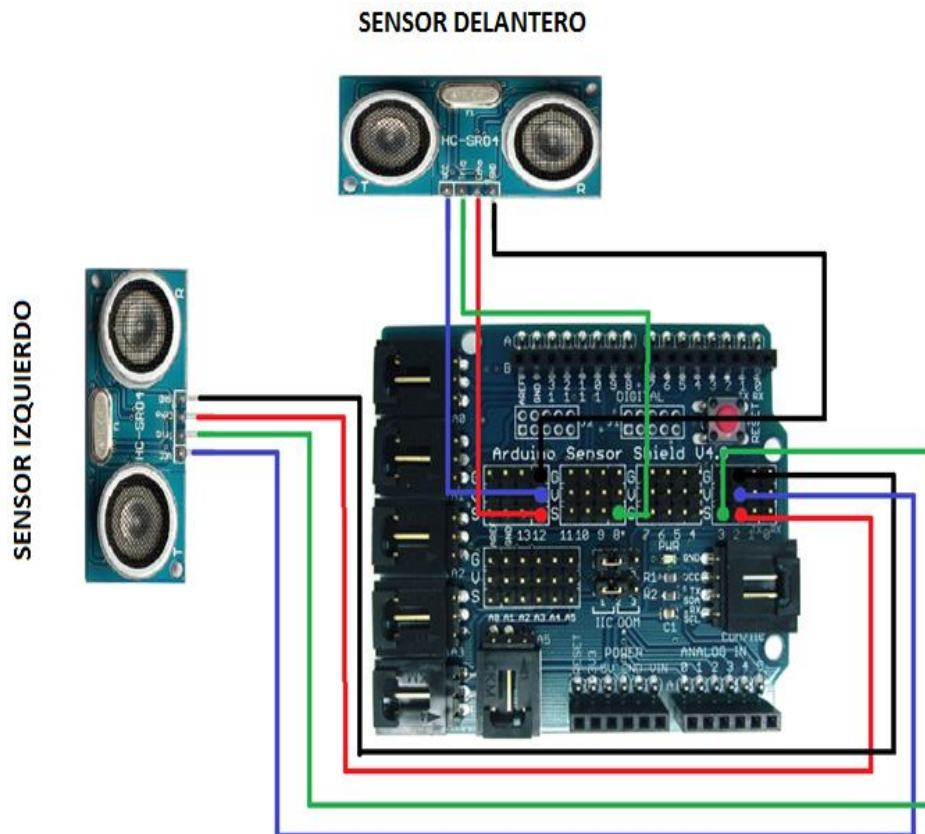


Figura 4.1: Conexionado de los sensores de ultrasonidos.

El programa que se desea crear, ha sido diseñado para que pueda cumplir las siguientes especificaciones que se muestran en el dibujo (Figura 4.2), ya que utilizando dichas medidas se puede garantizar que el robot siga correctamente las paredes de las habitaciones y además sea capaz de salir de una habitación y entrar en otra.

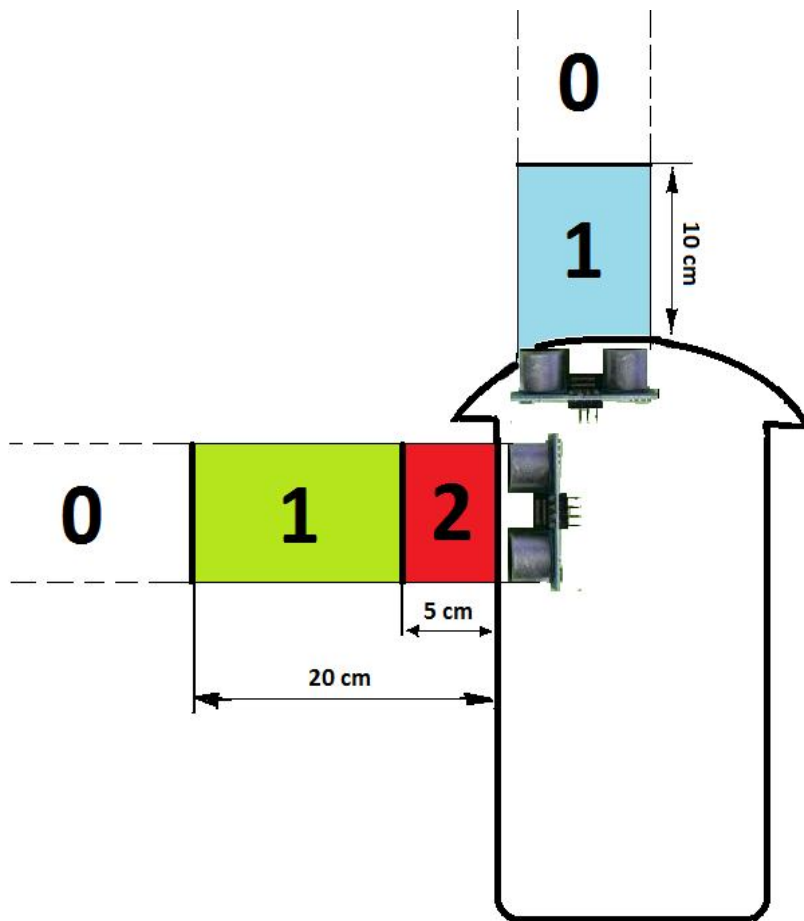


Figura 4.2: Rangos de distancias utilizados en la programación.

A continuación se muestra (Tabla 4.1) las posibles lecturas con los estados de los motores que se han elegido según dichas lecturas:

SENSOR IZQUIERDO	SENSOR DELANTERO	ESTADO MOTORES
2	1	GO RIGHT
2	0	ADELANTE Y DESVIACIÓN A LA DERECHA (separar de pared)
1	1	GO RIGHT
1	0	GO FORWARD
0	1	GO LEFT
0	0	GO LEFT+GO FORWARD

Tabla 4.1: Estado motores según medidas de los sensores.

4.1.2 Control de motores.

Se ha establecido la siguiente conexión realizada entre Arduino-Puente H-Motores (Figura 4.3), aunque cualquier otro conexionado con otras salidas digitales podría ser compatible siempre que se cumplan los requisitos de salidas PWM que se explicarán más adelante.

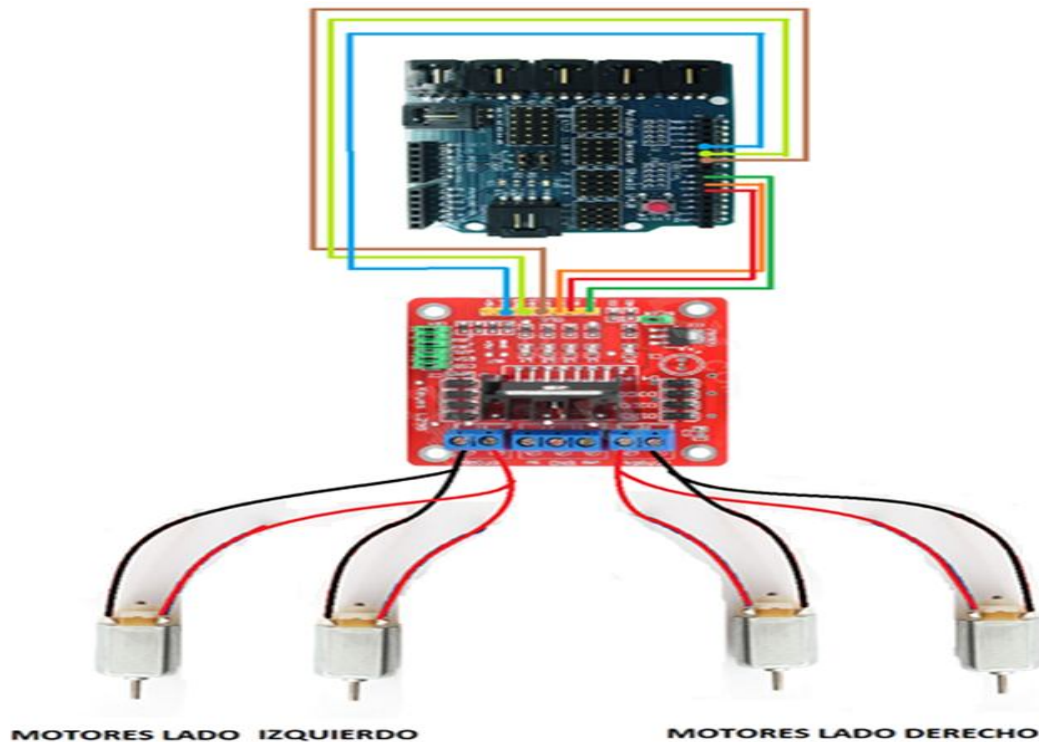


Figura 4.3: Conexión de motores-L298N-Sensor Shield de Arduino.

A continuación se muestra la lógica (Tabla 4.2) utilizada para el control de los motores:

EN A	EN 1	EN 2	EN 3	EN 4	EN B	ESTADO MOTORES
1	0	0	0	0	1	STOP
1	0	1	0	1	1	GO FORWARD
1	0	1	0	1	1	GO BACKWARD
1	0	1	1	0	1	GO LEFT
1	1	0	0	1	1	GO RIGHT

Tabla 4.2: Lógica de las salidas de Arduino según el estado de los motores.

Nomenclatura pines de entrada de la etapa de potencia del control de motores:

- EN A: Habilita el funcionamiento del motor A.
- EN 1 y EN 2: Son los dos pines para indicar el sentido del motor A.
- EN B: Habilita el funcionamiento del motor B.
- EN 3 y EN 4: Son los dos pines para indicar el sentido del motor B.

Según las medidas, anteriormente descritas sobre la distancia de los sensores de ultrasonidos a las paredes, se han establecido los siguientes estados de los motores (Tabla 4.3) para poder recorrer todas las paredes de las habitaciones.

VARIABLE EN EL PROGRAMA	DESCRIPCIÓN	ESTADO MOTORES					
r	DERECHA	GO RIGHT					
s	EVITAR CHOCAR CON PARED QUE SE ESTÁ SIGUIENDO (se da menos velocidad a ruedas derechas para hacer que se separe de la pared haciendo girar a la derecha ligeramente)	EN A	EN 1	EN 2	EN 3	EN 4	EN B
		1	0	100	0	255	1
		Además los pines digitales para el control de motores permiten la modulación PWM y se puede indicar el ciclo de trabajo para los motores entre 0-255 (0%-100%) funcionando como analógicos.					
f	SIGUE RECTO AL ESTAR A LA DISTANCIA CORRECTA DE LA PARED QUE SIGUE Y NO HAY OBSTÁCULOS DELANTE	GO FORWARD					
l	DELANTE HAY PARED Y A LA IZQUIERDA HAY CAMINO LIBRE: Gira a la izquierda	GO LEFT					
t	INTENTA BUSCAR LA PARED AL SEPARARSE DEMASIADO DE ELLA (gira a la izquierda y sigue un poco hacia delante)	GO LEFT (500 ms) + GO FORWARD (300 ms)					

Tabla 4.3: Variables para modificar el estado de los motores.

La programación de estos estados se ha implementado de la siguiente forma:

GO_RIGHT:

```
if ((medidaDelante==1)&&((medidalzquierda==2) || (medidalzquierda==1))) dato='r';
```

RECTO PERO DESVIÁNDOSE A LA DERECHA PARA NO CHOCAR:

```
if ((medidaDelante==0)&&(medidalzquierda==2)) dato='s';
```

GO_FORWARD:

```
if ((medidaDelante==0)&&(medidalzquierda==1)) dato='f';
```

GO_LEFT:

```
if ((medidaDelante==1)&&(medidalzquierda==0)) dato='l';
```

GO_LEFT+GO_FORWARD:

```
if ((medidaDelante==0)&&(medidalzquierda==0)) dato='t';
```

4.1.3 Flujograma de la programación del robot móvil.

A continuación se ha realizado un flujograma (Figura 4.4), de todo el programa realizado para el robot móvil para realizar el movimiento por las habitaciones de forma autónoma, y de esta forma comprender mejor el funcionamiento de éste.

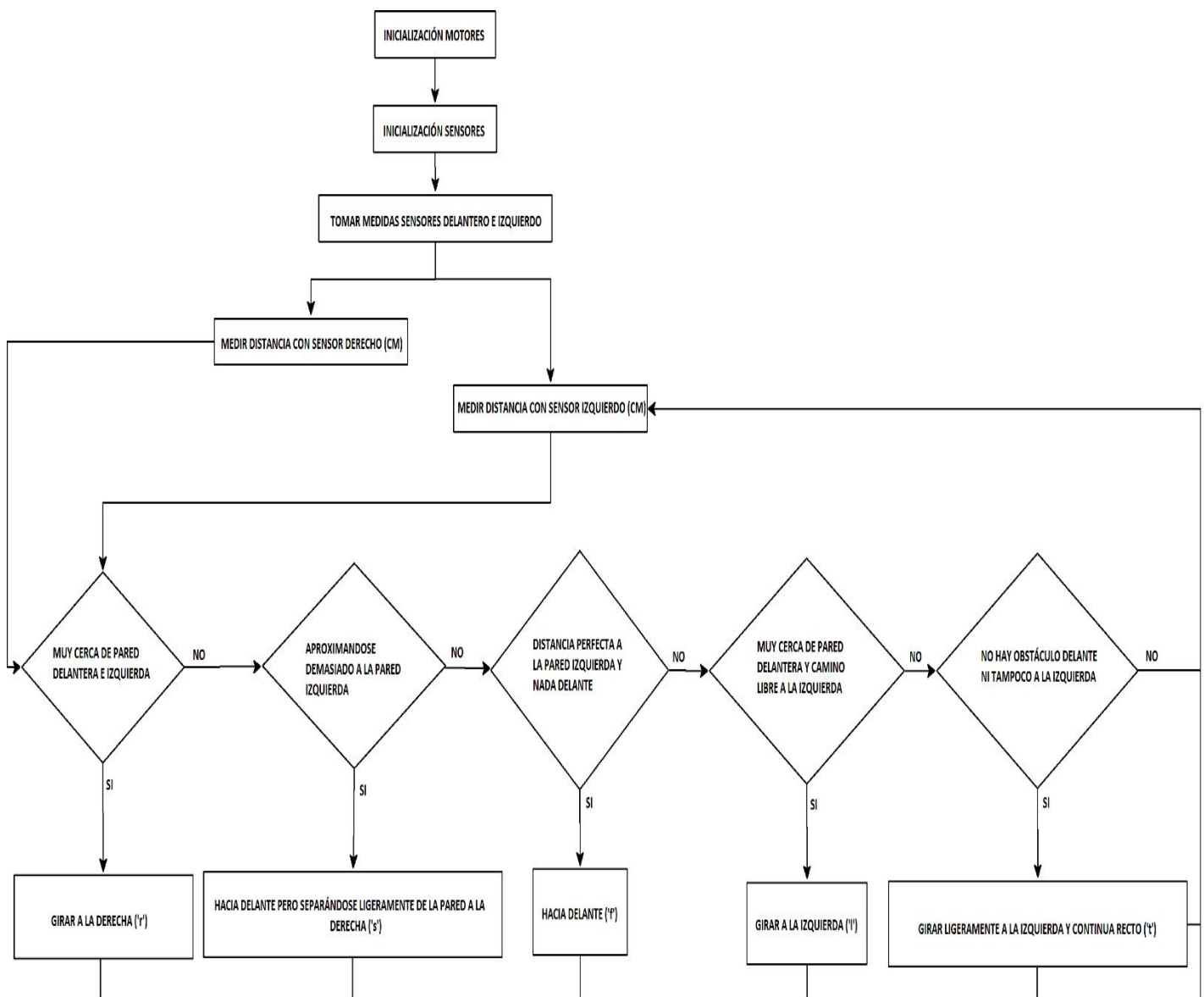


Figura 4.4: Flujograma del programa desarrollado para el movimiento del robot móvil.

4.2 Parte 2: Detección de personas y habitaciones.

En este apartado se explicará los pasos que se han realizado para configurar todos los dispositivos utilizados en la Raspberry Pi, instalación de librerías para comenzar con la programación y configuración del control remoto por SSH para utilizar la Raspberry Pi desde otro PC, por comodidad, para no utilizar teclado y monitor cada vez que se desee modificar algo en ella.

Por otro lado, se describirá la configuración y trabajo que realizará el PC externo para realizar la monitorización de la Raspberry Pi, así como la utilización de ROS y la programación de los diferentes nodos.

4.2.1 Instalación del Sistema Operativo en la Raspberry Pi.

Raspberry Pi permite instalar una gran variedad de Sistemas Operativos de la familia de Linux como Raspbian (Debian), Pidora (Fedora) o Arch Linux, entre otros.

Cualquiera de estos sistemas operativos sirven para realizar este proyecto, pero finalmente se optó por una versión de Raspbian cuya imagen ya incluía el software ROS que se necesitará más adelante cuya función se explicará a continuación.

También se podía haber optado por partir de una imagen de Raspbian limpia y haber instalado ROS desde cero, pero se encontró una imagen de Raspbian con la versión ROS Fuerte que funcionaba correctamente y se descartó instalarlo desde cero, ya que, por experiencia en otros PCs donde ya se había instalado ROS, el proceso de instalación requería bastante tiempo y si se cometía un fallo habría que volver a empezar por lo que hace innecesario instalar el sistema ROS en un Sistema Operativo limpio si ya existe una imagen de un SO con ROS funcionando.

Para realizar la instalación de la imagen Raspbian+ROS Fuerte en la tarjeta de memoria SD que se utilizará en la Raspberry Pi, se necesitó una SD con memoria mayor de 3,63 GB, que es lo que ocupa la imagen que se instalará en ella.

Una vez instalado el Sistema Operativo en la Raspberry Pi se pasó a configurar la comunicación por SSH y a instalar los drivers del adaptador WIFI Belkin descrito anteriormente para podernos comunicar por SSH y ROS desde nuestro ordenador a la Raspberry Pi y viceversa.

4.2.2 Configuración del control remoto de la Raspberry Pi.

La utilización del servidor SSH en la Raspberry Pi es muy importante en este proyecto porque de esta forma nos permite acceder remotamente desde otro PC y así poder configurar, programar y lanzar nuestros programas creados sin necesidad de tener una pantalla y un teclado conectados a la Raspberry Pi.

La configuración del SSH se realiza como en cualquier Sistema Operativo Linux por lo que no se explicará con más detalle.

De esta forma ya estaría configurada la Raspberry Pi para acceder remotamente desde otro ordenador a ella. Ahora lo único que faltaba era instalar los drivers del adaptador Wifi para poderse conectar remotamente a ella.

4.2.3 Configuración adaptador Wifi.

En nuestro caso al tratarse de un dispositivo Belkin se deben instalar dichos drivers, en caso de utilizar otro adaptador se deberán instalar sus propios drivers.

Para instalar los drivers del adaptador Belkin se lanzará el siguiente comando:

```
$ sudo apt-get install belkin-firmware
```

Una vez instalado ya se puede utilizar el adaptador Wifi.

Para realizar la conexión remota se debe configurar la conexión a la red a la que se quiere realizar la conexión ya sea de forma dinámica o estática. En este caso se ha configurado por comodidad una IP estática a la que se conectará la Raspberry Pi para conectarnos siempre a la misma IP cada vez que encendamos la Raspberry Pi y no tener que modificar las variables de entorno para ROS cada vez que se conecte a otra IP, pero esto es una configuración opcional que se ha realizado.

Una vez instalado y configurado todo se hace un reboot para que se conecte directamente a la red y ya se pueda controlar la Raspberry Pi por SSH desde cualquier PC.

4.2.4 Instalación de las librerías de OpenCV.

La detección de personas se realizará utilizando las librerías de OpenCV, por tanto se requiere de su instalación en la Raspberry Pi para su posterior utilización.

Para ello, desde la terminal de Raspbian, se introdujo el siguiente comando para instalar las librerías en Python de OpenCV:

```
$ sudo apt-get install python-opencv
```

De esta forma ya estarán instaladas las librerías y podremos empezar a realizar el programa de detección de rostros.

4.2.5 Librerías GPIO de la Raspberry Pi.

Como se ha explicado anteriormente, los pines de entrada de la GPIO van a servir para conectar el sensor de infrarrojos para detectar el color negro que se utilizará para saber en qué habitación se está accediendo.

Para la instalación de la librería GPIO se necesita antes instalar el archivo Python-dev, por lo que se instalará antes que el archivo para las GPIO:

```
$ sudo apt-get install python-dev
```

Para instalar la librería GPIO, desde la terminal descargamos el archivo y lo instalamos después de descomprimirlo:

```
$ wget 'http://downloads.sourceforge.net/project/raspberry-gpio-python/RPi.GPIO-0.5.4.tar.gz'
```

Ya estarían instaladas las librerías suficientes para poder empezar a utilizar el sensor infrarrojo que, a continuación, se va a explicar cómo se ha conectado a la GPIO y su utilización.

La GPIO de la Raspberry Pi model B o rev. 2, la cual es el modelo que se está utilizando, tiene la siguiente disposición de los pines (Figura 4.5).

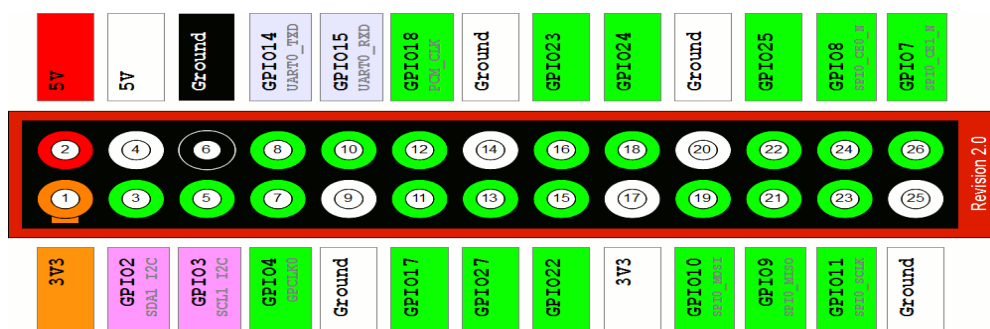


Figura 4.5: Pines de GPIO de la Raspberry Pi model B.

Los números que se pueden ver en los pines se corresponden a la nomenclatura del sistema GPIO.BOARD y el número que acompaña a las etiquetas que empiezan por "GPIO" se corresponden a la forma de nombrar los pines según la opción GPIO.BCM.

Esto se debe a que a la hora de programar los pines que tenemos conectados de las GPIO se debe asignar qué sistema utilizar para nombrar los pines porque varía de un sistema a otro.

En un principio, se trató de utilizar 2 sensores de infrarrojos (un sensor izquierdo y otro derecho) para detectar un código de barras de forma horizontal (Figura 4.6).





SENSOR IZQUIERDO	SENSOR DERECHO	Nº DE HABITACIÓN
		HABITACIÓN 1
		HABITACIÓN 2
		HABITACIÓN 3

Figura 4.6: Primer método de detección de habitaciones con 2 sensores infrarrojos.

Pero, con este método, se obtuvieron malos resultados, ya que era muy difícil que los 2 sensores accedieran al mismo tiempo al código de barras y además limitaba mucho el número de habitaciones que se podían detectar con 2 sensores realizando esta lectura horizontal.

Finalmente se decidió realizar la lectura del código de barras con un único sensor y de forma vertical, leyendo cada una de las líneas que se va encontrando al pasar sobre ellas como se explicará más adelante.

La forma que se ha elegido para conectar el sensor a los pines de la GPIO es la siguiente (Figura 4.7), aunque se puede elegir otra válida.

Se han conectado el pin de señal "S" del sensor al pin GPIO de entrada digital 3 del sistema GPIO.BCM, los pines de alimentación de los sensores "V+" a los pines de GPIO de 5V y la tierra de los sensores "G" a la tierra de la GPIO.

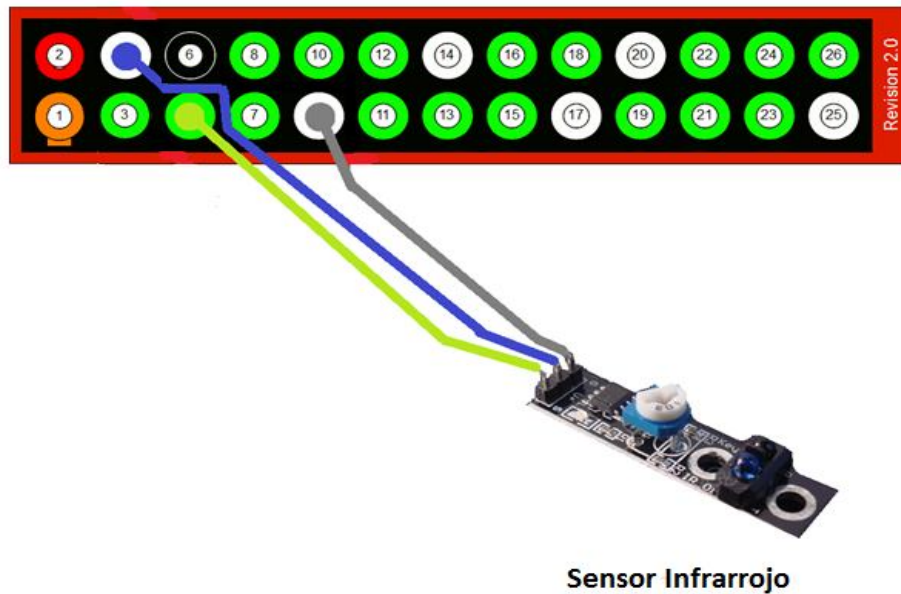


Figura 4.7: Esquema conexionado sensor infrarrojo.

Antes de lanzar el programa se debe ser usuario root para poder utilizar las GPIO, sino nos dirá que no tenemos permisos para utilizar la memoria de acceso a las GPIO de la Raspberry Pi.

Por esta razón a partir de ahora siempre accederemos a la Raspberry Pi como usuarios root para tener todos los permisos.

El programa en Python utilizado de testeo de los sensores y librería es el siguiente:

```
import RPi.GPIO as GPIO # Se importa la librería GPIO
GPIO.setmode(GPIO.BCM) # Modelo para nombrar los pines
GPIO.setup(3, GPIO.IN) ## GPIO 3 como entrada
while (true):
    sen=GPIO.input(3)
    print "Sensor:", sen
```

Tras comprobar que todo funciona correctamente seguimos con la configuración del resto.

4.2.6 Configuración del PC.

Para poder comunicarnos mediante ROS con la Raspberry Pi, y también realizar el control remoto por SSH, se va a utilizar el Sistema Operativo Ubuntu 12.04 pero se podría haber utilizado cualquier otro.

En este caso se ha decidido utilizar una máquina virtual con Ubuntu por temas de compatibilidad de ROS al ser el Sistema Operativo de la Raspberry Pi un Linux (Raspbian).

Se ha utilizado el programa de Oracle "VM Virtual Box" para utilizar este Linux de forma virtual porque el PC que se utiliza es un Windows 8.1 de 64bits con procesador Intel i5 de 4 núcleos a 2,53GHz y 4 GB de memoria RAM.

Dentro de la máquina virtual de Ubuntu cabe destacar que se ha establecido la conexión al router Wifi utilizado con una IP estática, para evitar que cada vez que nos conectemos a ella cambie de IP y tener que volver a configurar las variables de entorno de ROS que se explicarán más adelante y, además, se configuró la comunicación remota entre Raspberry Pi y PC mediante SSH, como ya se comentó antes, para prescindir de periféricos como ratón, teclado y monitor.

De aquí en adelante nos podremos conectar a la Raspberry Pi desde la máquina virtual de Ubuntu mediante una conexión de control remoto por SSH y realizar desde el PC los pasos siguientes que se siguieron.

4.2.7 Detección de personas.

Para la detección de personas se barajaron varias soluciones para ofrecer una solución al problema de ser capaz de detectar personas en una habitación.

La primera de ellas fue utilizar una cámara Kinect para detectar personas, ya que es la solución más eficiente y más fiable en las detecciones poder contar con detecciones en 3D, pero finalmente fue descartada por ser muy pesada la cámara para el robot móvil que se va a utilizar para el desplazamiento por las habitaciones. Por este motivo se decidió utilizar una webcam por ser mucho más ligera, de la que se disponía en concreto una cámara EyeToy como se ha descrito anteriormente.

La solución que se pensó y que a continuación se explicará, requiere del uso de las librerías OpenCV.

- **HAAR Cascade Classifier**

Se optó por utilizar este método para detectar personas, descrito anteriormente su funcionamiento en el capítulo de Estado del Arte, en el que según un clasificador en cascada HAAR dado se buscan las características contenidas de dicho clasificador en las imágenes o en este caso en los frames de la cámara para encontrar la parte del cuerpo que se quiere localizar.

Lo ideal hubiera sido crear un clasificador en cascada de tipo HAAR específico para este proyecto para detectar rostros, entrenándolo con imágenes positivas y negativas, pero existen clasificadores HAAR Cascade ya creados en OpenCV para la detección de partes del cuerpo de personas.

Principalmente para este proyecto se han probado los clasificadores para buscar: Rostros, parte inferior y parte superior del cuerpo.

- **HAAR CASCADE LOWER BODY**

En un principio se intentó detectar la parte inferior del cuerpo (pies, piernas y cintura) con este HAAR Cascade, pero los resultados que se obtuvieron no fueron buenos, ya que se requería detectar pies, piernas y cintura para obtener alguna detección, lo que requería tomar detecciones a mucha distancia (Figura 4.8), cerca de los 3 metros, para poder encuadrar dentro de la vista de la cámara la parte inferior del cuerpo de una persona incluidos los pies.



Figura 4.8: Detección con Haar Cascade Classifier Lower Body.

Esto hace este método poco eficiente porque solo detectaría personas que se encuentren lejos para que encuadre la cámara la mitad del cuerpo, cuando alguien esté más cerca del robot no se detectaría.

- HAAR CASCADE UPPER BODY

Luego se pensó en detectar la parte superior del cuerpo, con mejores resultados que el anterior, pero no del todo satisfactorias.

Este clasificador, básicamente, las principales características que incluía eran para detectar los hombros, pero no está muy optimizado porque es muy sensible al contraste con el fondo según el tipo de camiseta que lleve la persona.

Por ejemplo, si el borde del contorno del hombro no tenía mucho contraste con el fondo la detección no se realizaba (Figura 4.9).

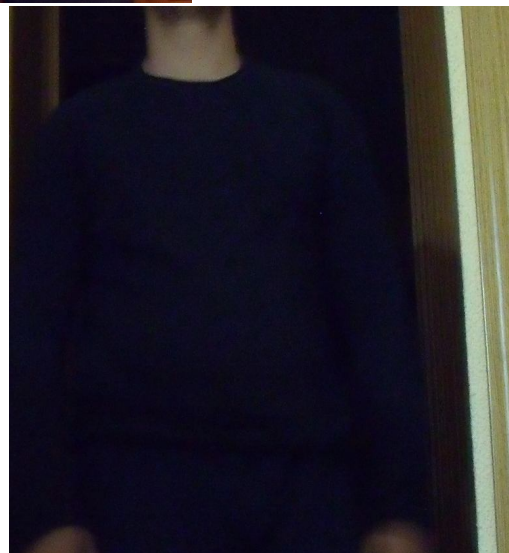
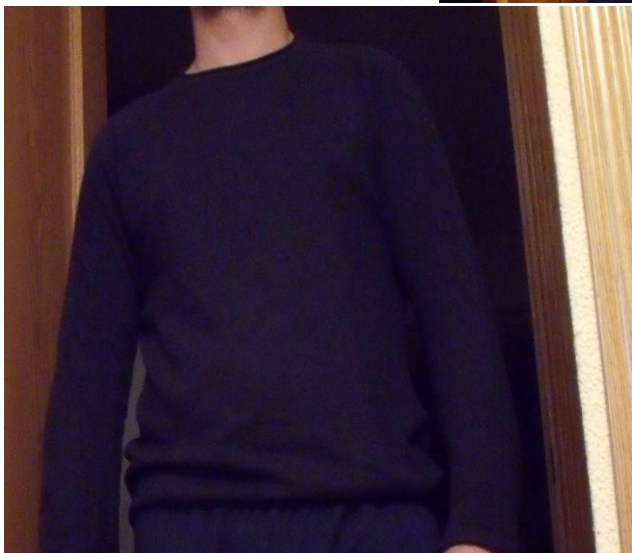


Figura 4.9: Detección utilizando Haar Cascade Classifier Upper Body (superior) y Fallos en la detección (inferiores).

Este método también se descartó por depender tanto el fondo para el resultado de la detección.

- HAAR CASCADE FRONTAL FACE

Después de descartar los clasificadores anteriores, se optó por probar el clasificador para detectar caras de frente.

Este clasificador está mucho más optimizado que los anteriores, ya que no se ha detectado que influya en las detecciones el fondo de la imagen.

Además en nuestro caso se va a utilizar la función **detectMultiScale** de OpenCV, la cual en cada iteración se va a disminuir la imagen un tanto por ciento según el factor de reducción que se le indique hasta llegar a un tamaño mínimo predefinido. Esta función lo que aporta es que para imágenes grandes se van disminuyendo su tamaño para que las subventanas de búsqueda de características del clasificador en cascada coincidan con los rostros de gran tamaño de las imágenes.

En cada iteración se detectarán varias caras aunque solo haya una sola cara en la imagen, esto es debido a la reducción de la imagen que también han pasado todos los nodos de características del clasificador en cascada.

Para ello se debe hacer una fusión de las detecciones (Figura 4.10) para garantizar que es una cara.

Primero se hace un dilatado y erosión para juntar las detecciones y a continuación selecciona la que más probabilidad tiene de ser una cara.

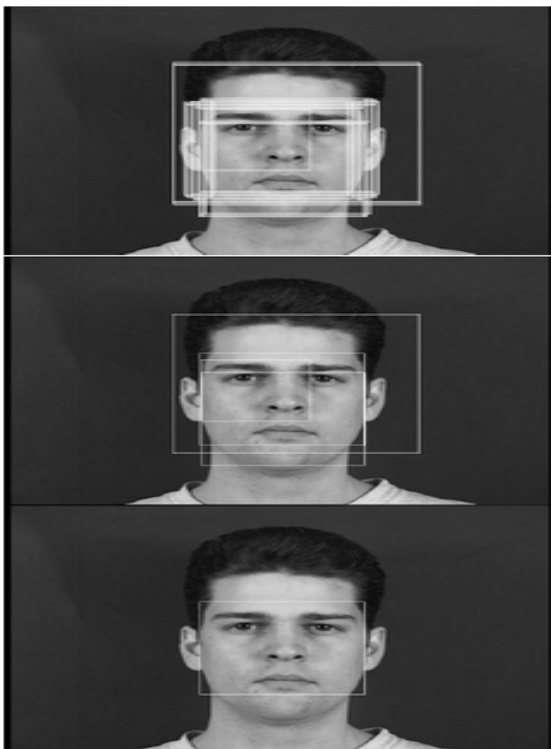


Figura 4.10: Fusión de las detecciones a varias escalas para verificar una cara.

La principal función que utiliza el programa para la detección de rostros es la mostrada a continuación, la cual se le indica el clasificador cascada utilizado, factor de

reducción de la imagen en cada iteración para la búsqueda de rostros y el número de vecinos mínimos que deben tener los rectángulos candidatos con estas características.

```
cascade = cv2.CascadeClassifier("haarcascade_frontalface_alt.xml")
```

```
rects = cascade.detectMultiScale(gray_frame, 1.2, 2)
```

Este método funciona perfectamente pero al estar situada la cámara tan abajo por ser el robot tan pequeño se pierde alguna detección de rostro. Para intentar solucionar esto se decidió elevar la cámara un poco para tener mayor ángulo de visión.

Por lo que se ha utilizado, como se describía anteriormente, una vieja bobina de DVDs alta, para elevar la cámara y conseguir orientarla lo suficiente para que las detecciones sean mejores, consiguiendo aumentar la altura 20 cm más, haciendo un total de 31,5 cm desde el suelo a la posición de la cámara.

Cuanto más se consiga levantar la cámara a la altura de una persona mejor serán las detecciones porque el clasificador de cascada usado ha sido entrenado con caras de vista frontal, por lo que mejores resultados obtendremos si se eleva la cámara a la altura de la cara.

Por último, cabe destacar que para disminuir el tiempo que se tomaba para realizar la búsqueda de rostros en cada frame se decidió reducir a la mitad el área donde se realizará la búsqueda de los rostros en cada frame (Figura 4.11), de esta forma no se tienen que buscar rostros en toda la imagen que captura la cámara en cada frame, ya que es más probable encontrar rostros en la mitad superior de las imágenes al estar situada la cámara a una altura baja.

Esto fue necesario realizarlo debido a que se obtenían "Timeouts" al requerir más tiempo en buscar rostros en toda la imagen.

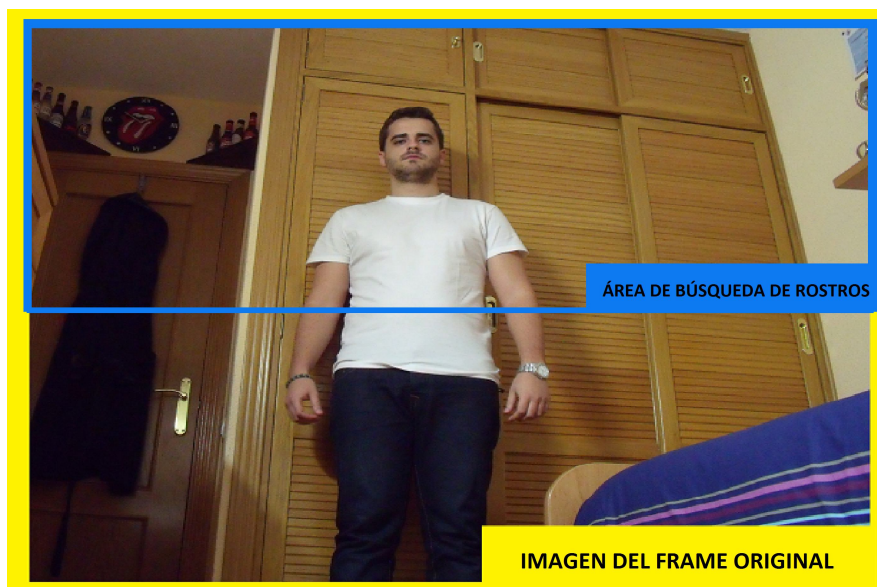


Figura 4.11: Área de búsqueda de rostros respecto a la imagen completa de cada "frame".

4.2.8 Utilización de ROS para la monitorización.

Una vez se terminó de instalar, configurar y testear que todo funcionaba bien por separado (detección de rostros y lectura del sensor infrarrojo), se pasó a implementarlo dentro de ROS para hacer todas las funciones a la vez.

Lo que se quería hacer desde un primer momento era quitar coste computacional a la Raspberry Pi, ya que posee un procesador de 700 MHZ y una memoria RAM de 512 MB suficientes pero muy justa para el procesamiento de imágenes, que en algunos casos mientras se realizaba la detección de rostros, usando el HAAR Cascade anteriormente mencionado, se obtenía algún Timeout debido al tiempo empleado en la detección de rostros.

Por ello, no se quiere dar más operaciones complejas al sistema, por eso, se decidió tomar las medidas del sensor y el número de rostros detectados y mandarlos a un PC para que realice todas las operaciones restantes.

En primer lugar, se pensó en mandar dichas medidas a un ordenador utilizando SSH, ya que es un proceso sencillo, pero luego se decidió utilizar ROS que permite crear un MASTER, el PC, y un HOST, en la Raspberry Pi, y realiza la comunicación entre ellos enviando "topics" a los nodos creados en cada uno de ellos.

Además, utilizar este método, es muy útil para posibles futuros proyectos tener ya instalado ROS en la Raspberry Pi del robot móvil porque permite muchas funcionalidades.

4.2.9 Instalación de ROS.

Como se indicó antes, la imagen utilizada de Raspbian ya traía instalado ROS con la versión Fuerte, por lo que, no es necesario instalar ROS en ella.

Por el contrario, para realizar la conexión y envío de mensajes entre la Raspberry Pi y el PC se necesita instalar en el PC el sistema ROS.

Se decidió instalar la misma versión de ROS Fuerte que es la que está instalada en la Raspberry Pi para evitar posibles incompatibilidades.

- **PROGRAMACIÓN DE LOS NODOS:**

El gráfico (Figura 4.12) resume que la Raspberry Pi realiza la detección de rostros y la lectura del estado del sensor infrarrojo que se activa con el color negro. Estas detecciones se enviarán por 2 topics al ROSCORE situado en el PC.

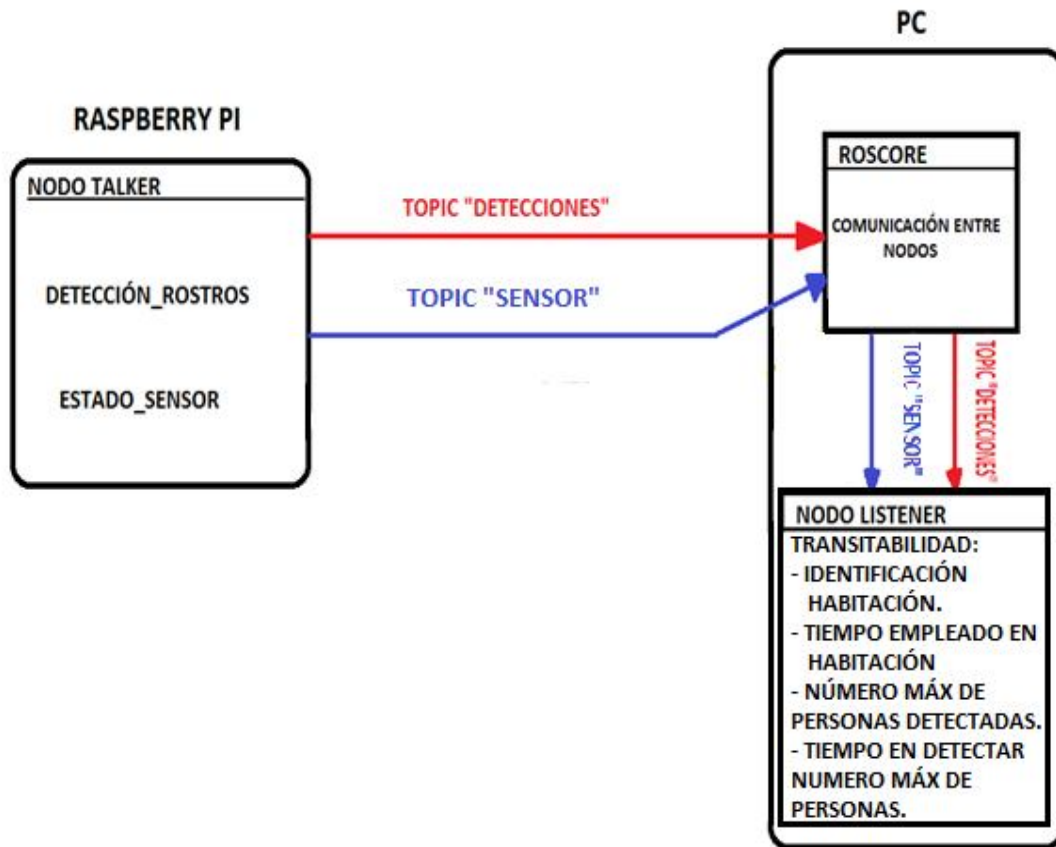


Figura 4.12: Funcionamiento del sistema ROS planteado.

ROSCORE es el encargado de la comunicación entre los nodos del sistema, entre otras funciones, por lo que una vez recibidos los topics los envía al nodo Listener para que se encargue de analizar los datos y poder saber las detecciones, tiempo transcurrido en detectar todas las personas y tiempo empleado en cada una de las habitaciones por las que pasa el robot.

▪ NODO TALKER

El programa que se ha realizado para crear el nodo Talker (Figura 4.13) se encuentra situado en la Raspberry Pi.

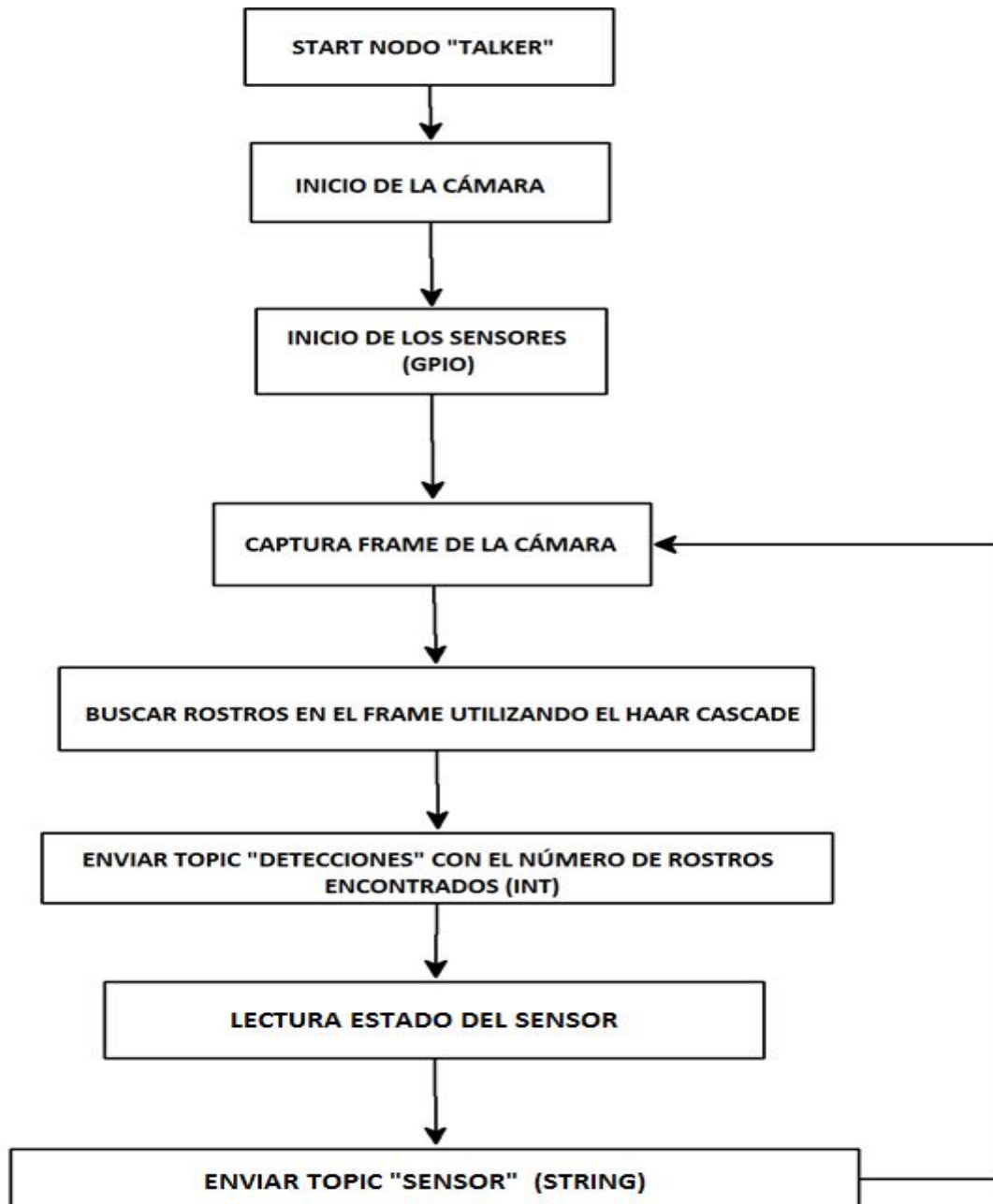


Figura 4.13: Flujograma del Nodo Talker utilizado en la Raspberry Pi.

Los datos son mandados por 2 topics al nodo Listener del PC:

- Topic "Detecciones": Envía un entero (integer) con el número de rostros que ha detectado en cada momento.
- Topic "Sensor": Envía una cadena de caracteres (string) con el estado del sensor infrarrojo derecho.

"False" -> No detecta color Negro. "True" -> Si ha detectado color Negro.

▪ NODO LISTENER

Dicho nodo se encarga de recopilar los datos y además realiza toda la lógica para saber la habitación en la que se encuentra y el máximo número de rostros que se ha conseguido detectar (Figura 4.14).

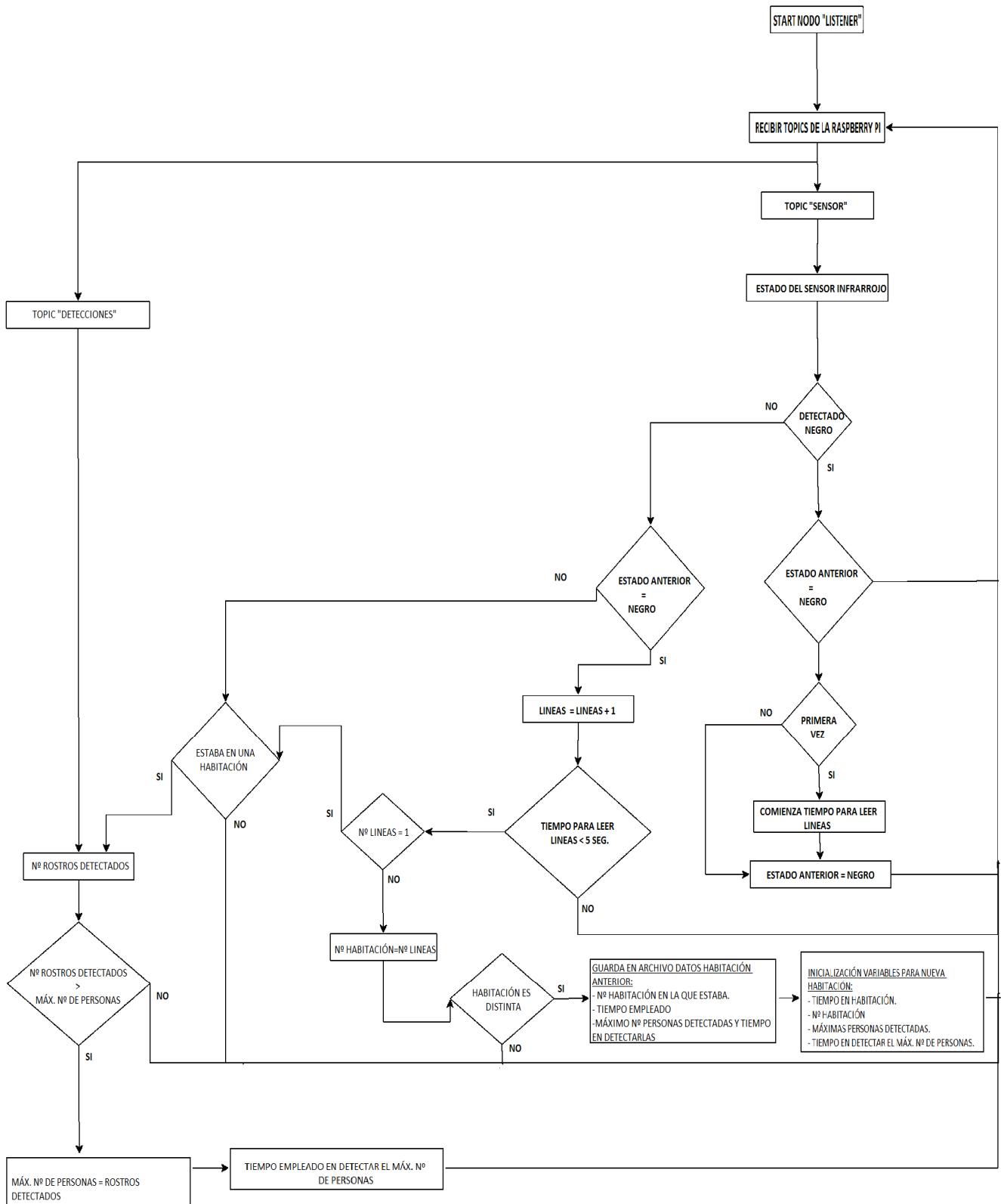


Figura 4.14: Flujograma del Nodo Listener utilizado en el PC para la monitorización de los datos.

También se realizará un “back up” cada vez que se cambie de habitación para, de esta forma, tener guardado en un archivo las habitaciones por las que ha pasado el robot, el número máximo de rostros detectados, tiempo en detectar máximo de rostros y tiempo empleado en recorrer las habitaciones.

Los códigos utilizados para detectar la habitación por la que pasa el robot se asemeja a un código de barras. Al utilizar un único sensor la lectura de las líneas negras las irá leyendo según pase sobre ellas. El tiempo máximo que se le da al sistema para leer el código de cada habitación se ha considerado de 5 segundos, debido a que se van a leer como máximo 4 líneas negras para 3 habitaciones que se van a considerar como máximo para este proyecto. Esta suposición de 3 habitaciones ha sido tomada para seguir un criterio y establecer esta codificación pero pueden ser ampliadas a más habitaciones y hacer una codificación personalizada para las necesidades de cada lugar en el que se vaya a utilizar sabiendo el número de habitaciones, ya que el número de habitaciones no es un problema en este proyecto.

En este caso la codificación utilizada para cada habitación es la siguiente:



- Si solo se ha detectado una línea se ignorará y se seguirá en la habitación en la que estaba antes de detectar la línea. Esto se hace para evitar falsos positivos.



- Si se detectan dos líneas en esos 5 segundos, estará en la habitación número 1.



- Si se detectan tres líneas en esos 5 segundos, estará en la habitación número 2.



- Si se detectan cuatro líneas en esos 5 segundos, estará en la habitación número 3.

Para hacer esta comunicación entre nodos posible, se necesita además configurar las variables de entorno en la Raspberry Pi y el PC para que el ROSCORE sepa quién es el MASTER y quién el HOST.

4.2.10 Configuración de las variables de entorno de ROS.

Las variables de entorno para la Raspberry Pi se van a configurar en el usuario "root" porque va a ser desde el cuál se van a lanzar los programas ya que las GPIO de Raspberry Pi, como se ha dicho anteriormente, necesitan permisos de root para poder trabajar con la librería de GPIO.

`$ export ROS_MASTER_URI= IP del MASTER (PC) -> Indica donde se encuentra el MASTER para mandar los topics`

`$ export ROS_HOSTNAME=(IP desde donde se realizará el envío) -> HOST, en este caso situado en la Raspberry Pi.`

Para las variables de entorno en el PC se debe realizar el mismo procedimiento que antes, pero se debe poner la misma dirección IP del PC para el MASTER como para el HOST por ser el PC el MASTER y además de ser, en este caso, donde van a ir a parar las transferencias de mensajes al nodo LISTENER.

En este caso no hace falta ser root para utilizar el nodo LISTENER porque el programa no necesita permisos especiales para las librerías ni funciones utilizadas en él.

De esta forma, está todo preparado para poder comenzar a utilizar ROS con comunicación entre los nodos.

4.2.11 Inicialización del sistema de monitorización Raspberry Pi-PC.

Para comenzar el sistema de detección de rostros por habitaciones, se deben seguir los siguientes pasos para poder conectar todo correctamente:

- **INICIALIZACIÓN DEL ROSCORE**

Antes de lanzar cualquier programa con ROS se debe tener un ROSCORE corriendo para que establezca las comunicaciones entre los nodos.

Para ello, se debe lanzar el ROSCORE desde el PC, ya que es donde hemos indicado que se encuentra el MASTER al que se le enviarán los datos recogidos por la Raspberry Pi.

Por tanto, abrimos una terminal en el PC y escribimos:

```
$ roscore
```

Se encarga de realizar las conexiones internas de las IPs dadas como variables de entorno.

Cuando haya cargado el ROSCORE correctamente ya se pueden lanzar los nodos para comenzar las detecciones (Nodo Talker) y monitorización de todos los datos (Nodo Listener).

En ningún momento se debe cerrar o parar la consola donde está corriendo ROSCORE porque pararía la comunicación de los nodos.

- **PUESTA EN MARCHA DE LOS NODOS**

Para comenzar con la monitorización de la Raspberry Pi en el ordenador se debe lanzar el nodo Listener, desde una nueva consola del PC (sin cerrar la consola de ROSCORE), poniendo:

```
$ rosrun [paquete_que_contiene_al_nodo] [nombre_del_archivo_del_nodo]
```

roslun -> comando de ROS para arrancar los nodos programados. Por lo tanto, se le debe indicar en qué paquete está el nodo y el nodo que queremos lanzar (listener.py).

Con este comando arrancamos el nodo Listener que está esperando que le lleguen datos de los topics enviados del nodo Talker.

Ahora debemos arrancar el nodo Talker desde la Raspberry Pi. Para ello, se debe encender la Raspberry Pi y conectarnos a ella por SSH desde otra consola del PC.

Una vez se ha accedido a ella remotamente, accedemos como usuario root para tener todos los permisos como se indicó antes.

Ahora lanzamos el nodo Talker de la misma forma que se indicó para el nodo anterior.

Finalmente, ya estaría todo listo y comenzará la monitorización en el nodo Listener de todos los datos que está tomando el nodo Talker situado en la Raspberry Pi.

5. EXPERIMENTOS

A continuación se van a describir los experimentos realizados y los resultados dados para intentar dar una idea de las prestaciones que tiene este sistema de reconocimiento de caras sobre una plataforma móvil.

5.1 Altura de la cámara.

Al utilizar una cámara para proyectos de robótica es muy importante saber en qué posición colocarla para obtener los mejores resultados. Con este sistema de detección se va a comprobar a qué distancias es capaz de detectar la cámara dependiendo de la altura.

Para la realización de estos experimentos se ha colocado la cámara en 3 alturas diferentes con el robot parado (está parado porque únicamente se ha querido comprobar a qué distancias es capaz de detectar el sistema de detección de rostros) y se han utilizado dos sujetos diferentes para comprobarlo, una persona de 1,75 metros de altura y la otra un niño de 1,25 metros. De esta forma se ha comprobado las distancias a las que detectará la cámara tanto para personas adultas como niños.

- Cámara a baja altura:

La cámara se situó a 14,3 cm del suelo (encima de la batería la cual se ha situado sobre el chasis del robot) además la cámara tiene una inclinación de 35° sobre el nivel del suelo para así apuntar con mejor ángulo hacia los rostros.

Los rangos en los que se han podido detectar rostros a esta altura son desde 100 cm hasta 190 cm con una persona de 1,25 metros de altura (Figura 5.1) y desde 170 cm hasta 200 cm con una persona de 1,75 metros de altura (Figura 5.2).

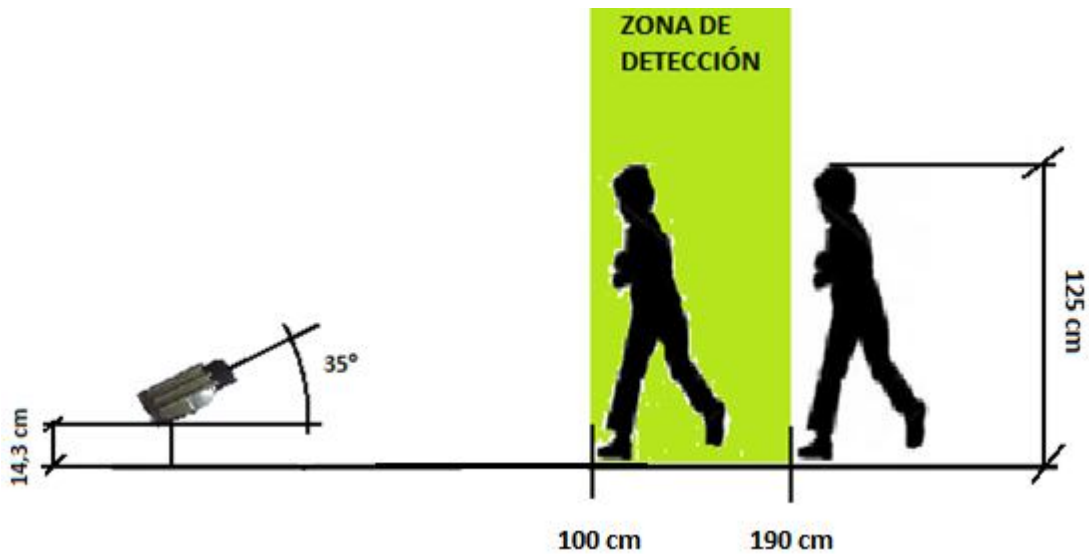


Figura 5.1: Distancia de detección con cámara a baja altura y personas de 1,25 metros de altura.

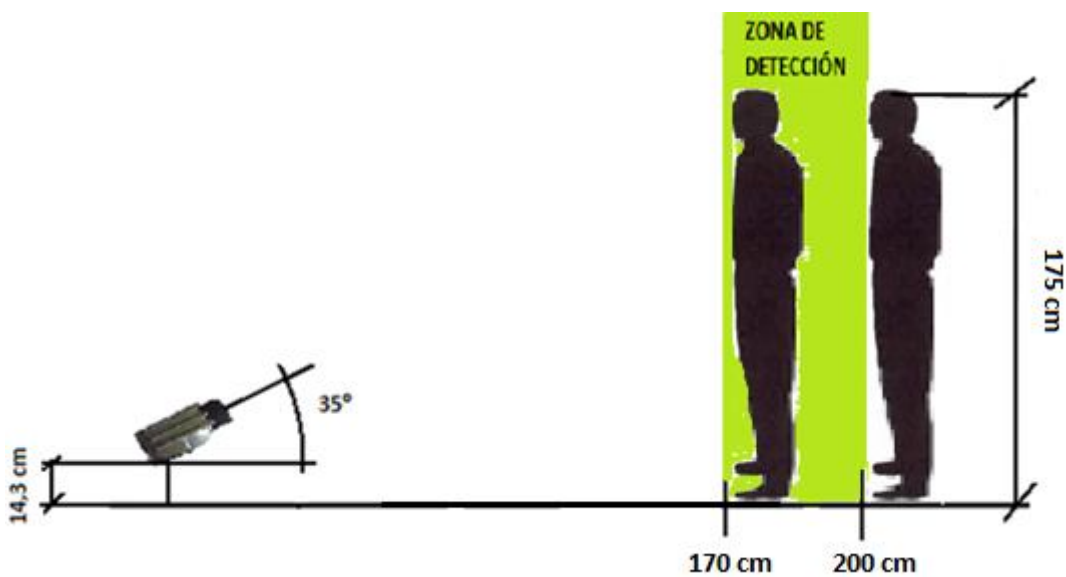


Figura 5.2: Distancia de detección con cámara a baja altura y personas de 1,75 metros de altura.

- Cámara a media altura:

Para esta prueba se situó la cámara en la posición que se ha indicado a lo largo de la memoria, encima de una bobina de DVDs situada sobre el robot. Por tanto, la cámara, está situada a una altura de 31,5 cm del suelo, con una inclinación de la cámara de 35° al igual que antes.

En este caso los rangos de distancia entre el robot y la persona en los que se ha conseguido detectar rostros son desde los 70 cm hasta los 140 cm en el caso de la persona de 1,25 m de altura (Figura 5.3) y desde 105 cm hasta los 150 cm en el caso de la persona adulta (Figura 5.4).

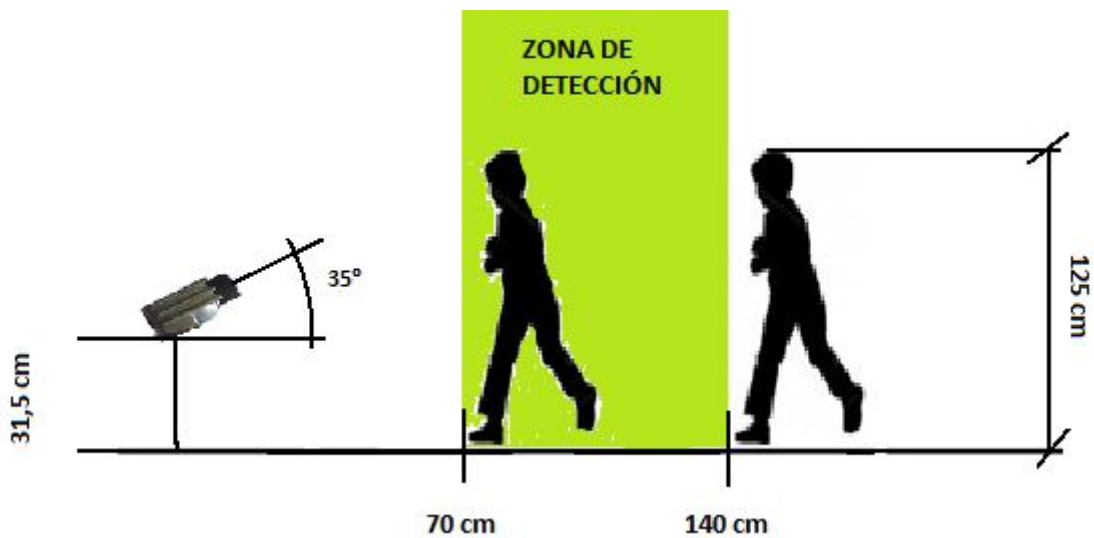


Figura 5.3: Distancia de detección con cámara a media altura y personas de 1,25 metros de altura.

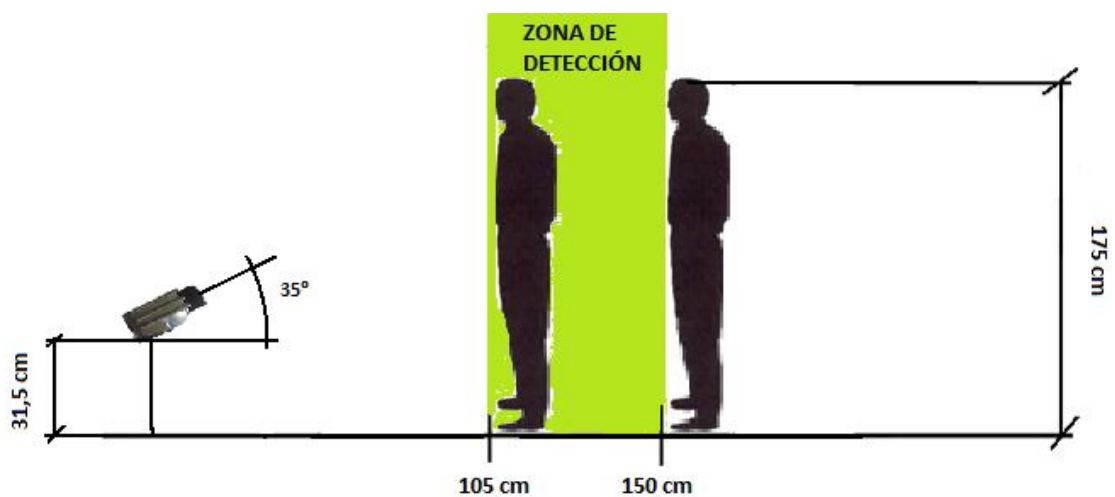


Figura 5.4: Distancia de detección con cámara a media altura y personas de 1,75 metros de altura.

- Cámara a altura elevada:

En este caso la cámara se ha situado sobre una mesa de 90 cm de altura sin inclinación de la cámara ya que no es necesario orientarla más hacia los rostros.

Los rangos de distancia a la que se pueden detectar rostros desde esta altura de la cámara son desde los 0 cm hasta los 50 cm en el caso de personas de 1,25 m de altura (Figura 5.5) y desde los 65 cm hasta los 190 cm para personas de 1,75 metros de altura (Figura 5.6).

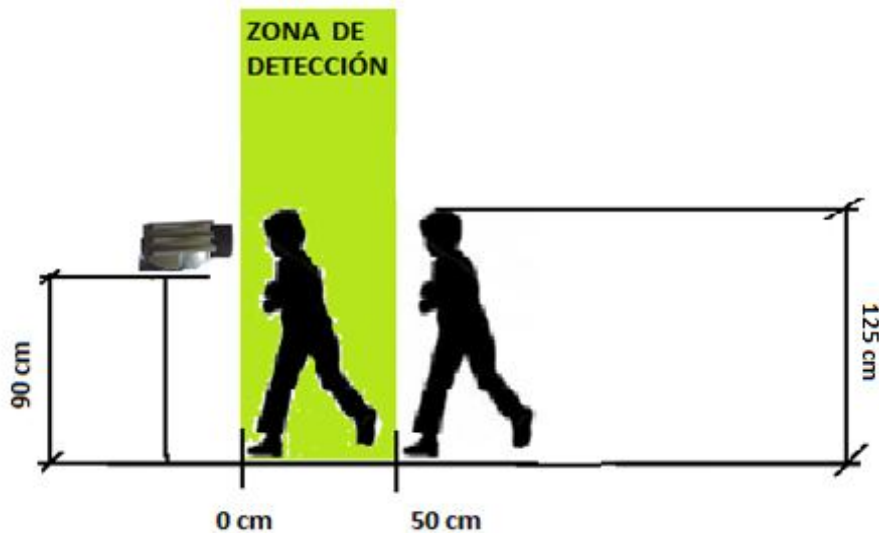


Figura 5.5: Distancia de detección con cámara a altura elevada y personas de 1,25 metros de altura.

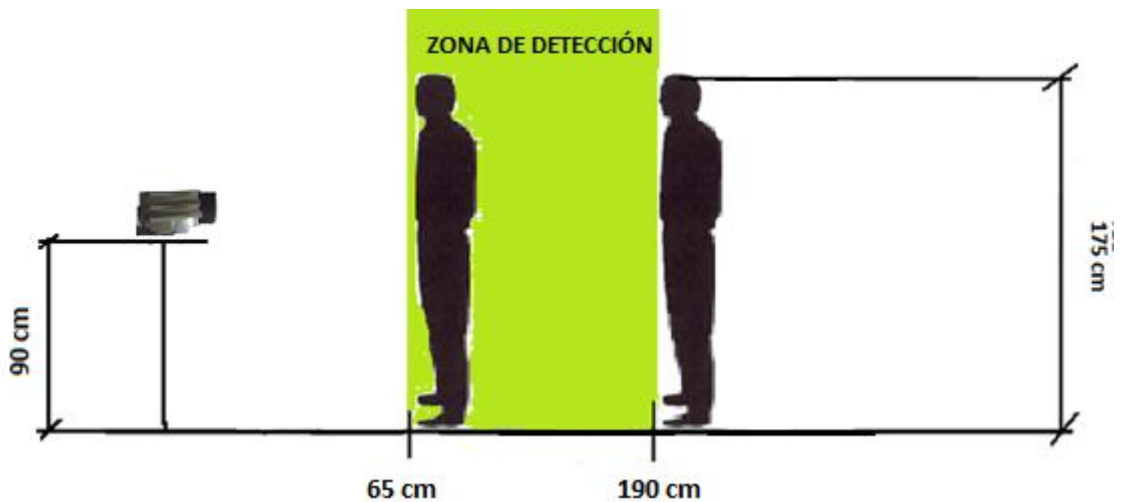


Figura 5.6: Distancia de detección con cámara a altura elevada y personas de 1,75 metros de altura.

Con estas pruebas se puede apreciar que las mejores detecciones se realizan cuanto más se acerque la altura de la cámara al de las personas que se quieren detectar. Por ello se utilizó para este proyecto una altura media de 31,5 cm, como se explicó en capítulos anteriores, ya que para este proyecto es suficiente esa altura para las

habitaciones en las que se realizaron las pruebas al ser su rango de detección a esa altura de la cámara suficiente.

Dependiendo de la aplicación y el tipo de personas que se quieren detectar es recomendable graduar la altura de la cámara para mejorar las detecciones.

5.2 Cambios de iluminación.

Se han realizado pruebas de detección de rostros en varios escenarios con iluminaciones diferentes con el fin de ver cómo afecta la iluminación al sistema de detección de rostros.

Para probar este problema del cambio de iluminación se han elegido únicamente 3 escenarios de iluminación con 1 persona en movimiento por la zona estudiada frente a la cámara del robot, ya que obviamente el escenario de iluminación baja (luces apagadas) no se han realizado experimentos porque ya se sabe que no va a funcionar al no tener el robot una cámara de visión nocturna o flash.

Las situaciones que se han elegido para comprobar la influencia de la iluminación en las detecciones son las siguientes:

- Iluminación artificial desde el techo.

Para la realización de este experimento se ha puesto al robot a dar vueltas en una habitación de 10 m² únicamente con una iluminación artificial que proviene del techo. La persona realizó movimientos frente al robot a distancias de 1 y 1,5 metros (Figura 5.7).

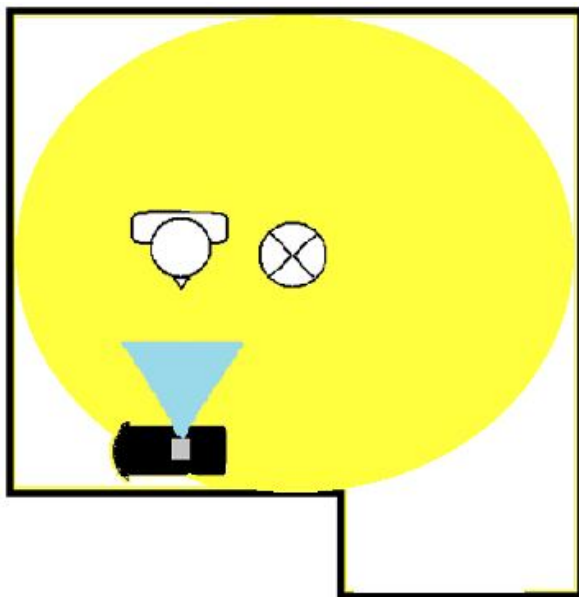


Figura 5.7: Momento en que la persona fue detectada por primera vez en habitación con iluminación artificial.

En todo momento la persona se estuvo moviendo frente al robot.

Los resultados obtenidos en esta situación de estudio fueron:

- Primera Prueba:

Tiempo transcurrido para la primera detección de rostro= 48,44 segundos.

El mismo rostro fue detectado 41 veces en 204 segundos que duró la prueba.

- Segunda Prueba:

Tiempo transcurrido para la primera detección de rostro= 55,8 segundos.

El mismo rostro fue detectado 33 veces en 262 segundos que duró la prueba.

- Iluminación natural favorable a las detecciones (luz de cara a la persona detectada).

En este experimento se ha seleccionado una habitación grande de 30 m² con ventanas en 2 de las 4 paredes por donde entra la luz del exterior.

Las pruebas para este experimento se han realizado mientras el robot recorría las paredes con las ventanas (luz proveniente tras la cámara del robot) tomando medidas en todas las zonas bien iluminadas.

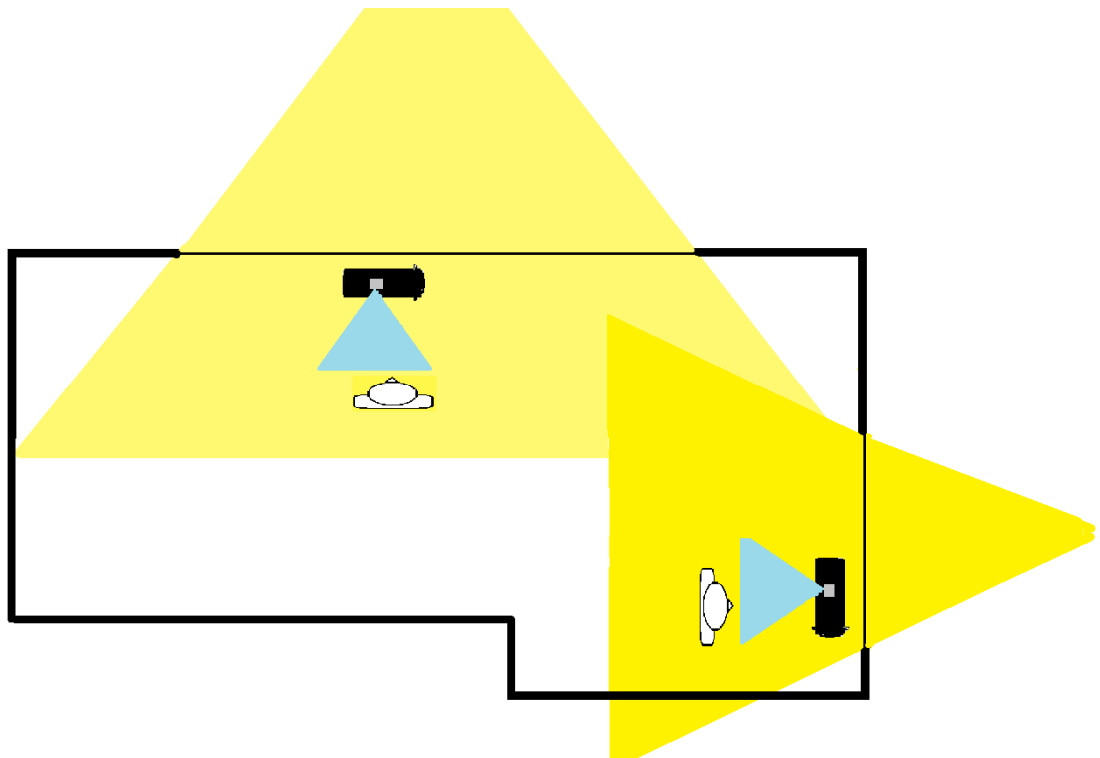


Figura 5.8: Momentos en que la persona era detectada en habitación con iluminación artificial.

La persona se movía delante del robot entre distancias de 1 y 1,5 metros y prácticamente detectó a esta persona cada vez que miraba de frente a la cámara del robot (Figura 5.8), ya que realizó detecciones cómo:

- Primera Prueba:

Tiempo en detectar el rostro por primera vez = 10,2 segundos.

Número total de veces que detectó el mismo rostro =90.

Tiempo total del experimento= 135,62 segundos.

- Segunda Prueba:

Tiempo en detectar el rostro por primera vez = 7 segundos.

Número total de veces que detectó el mismo rostro =104.

Tiempo total del experimento= 161,7 segundos.

- Tercera Prueba:

Tiempo en detectar el rostro por primera vez = 13 segundos.

Número total de veces que detectó el mismo rostro =77.

Tiempo total del experimento= 152,8 segundos.

Cada vez que se pasaba por delante de la cámara mirando de frente en dirección al robot se detectaba la cara.

Se puede ver claramente en el archivo guardado (Figura 5.9) que se genera tras hacer las detecciones cómo se distingue la persona cuando pasa por delante de la cámara. Se aprecian tramos de 1 (cuando se ha detectado el rostro) que se corresponden cada vez que se ha pasado por delante de la cámara.

Sensor	Detecciones
0	0
0	0
0	0
0	0
0	1
0	1
0	0
0	0
0	1
0	1
0	1
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	1
0	1
0	1
0	1
0	1
0	1
0	1
0	1
0	1
0	1
0	1
0	1
0	1
0	1
0	1
0	1
0	1
0	0
0	0

Figura 5.9: Archivo con claras detecciones de cuando se detectaba rostro de la persona al pasar por delante de la cámara.

- Iluminación desfavorable para las detecciones (cámara a contraluz).

Para realizar este experimento se ha utilizado la misma habitación que para el experimento anterior pero, para este caso, se han tomado medidas mientras el robot recorre las paredes que no hay ventanas (a contraluz).

En este caso las detecciones tomadas por la cámara quedan siempre a contraluz, ya que la luz que entra por las ventanas incide sobre el robot y deja oscurecida a la persona detectada (Figura 5.10).

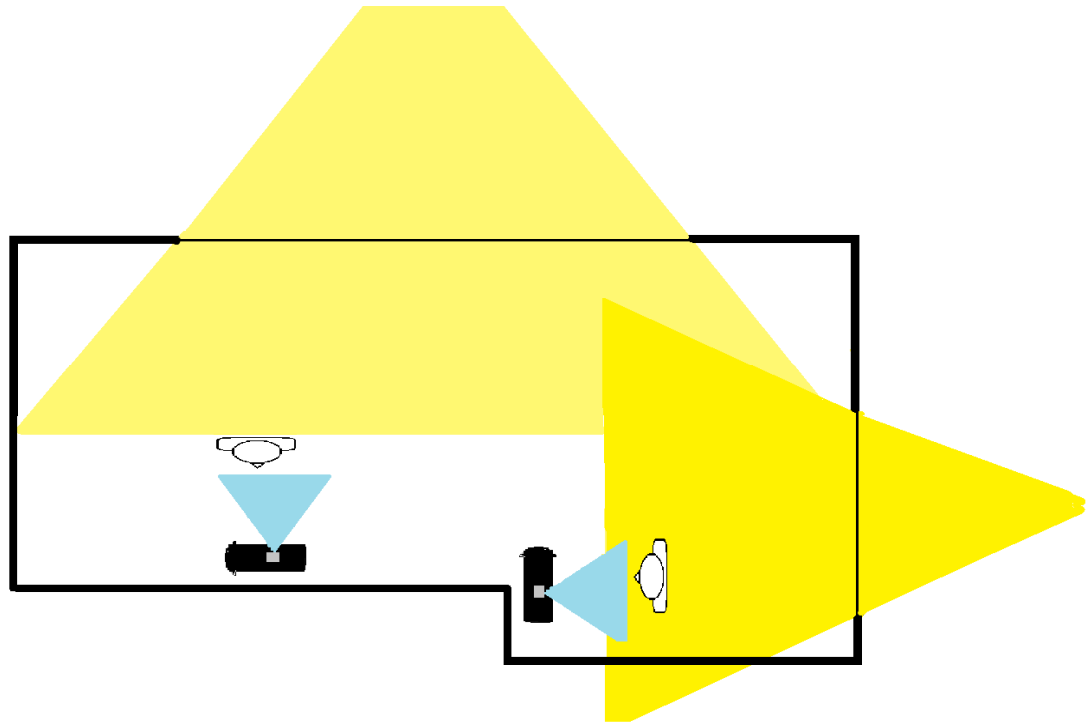


Figura 5.10: Momentos muy desfavorables para la detección de la persona a contra luz.

Las medidas que se han tomado con este experimento son las siguientes:

- Primera Prueba:

Tiempo en detectar el rostro por primera vez = 43,55 segundos.

Número total de veces que detectó el mismo rostro =21.

Tiempo total del experimento= 342 segundos.

- Segunda Prueba:

Tiempo en detectar el rostro por primera vez = 61 segundos.

Número total de veces que detectó el mismo rostro =25.

Tiempo total del experimento= 320 segundos.

- Tercera Prueba:

Tiempo en detectar el rostro por primera vez = 78 segundos.

Número total de veces que detectó el mismo rostro =36.

Tiempo total del experimento= 332 segundos.

Con estos experimentos en iluminación se puede ver que para iluminaciones favorables para la detección de rostros, con luz de cara a los rostros, se obtienen muy buenos resultados, detectándose prácticamente en todos los casos que el rostro se

encuentra frente a la cámara. También se obtienen buenos resultados con iluminaciones artificiales, en cambio con baja iluminación o casos desfavorables como detecciones a contra luz se obtienen bajas detecciones aunque se siguen detectando rostros a pesar de ello.

A continuación se van a realizar una serie de experimentos para saber cuánto tiempo tarda el sistema de detección de personas en detectar rostros dentro de la habitación. Se sabe que las medidas que se obtengan con estos experimentos no van a ser siempre los mismos, son unas medidas orientativas y lo que se quiere demostrar es el tiempo que ha tardado en detectar personas en estas habitaciones y con estas personas ya que en otras circunstancias como poca iluminación, personas que se mueven muy rápido o personas que no miran hacia el robot, se obtendrán peores resultados o en caso de no mirar hacia el robot no detectar rostros. Los experimentos que se han realizado para dar una idea del sistema de detección de personas creado en este proyecto son los siguientes.

5.3 Detección de 1 persona en distintas habitaciones.

En este caso se han realizado pruebas en 2 tipos de habitaciones, una habitación pequeña de 10 m² y en una grande de 30 m².

Además el sujeto al que se utilizó para las pruebas es una persona de 1,75 m de altura, a la que se le realizaron dos tipos de experimentos para su detección.

Uno de ellos es detectar a esta persona dentro de las habitaciones estando quieto en una posición fija en todo momento y el otro es detectar a la persona estando en movimiento por la habitación.

A continuación se muestran los resultados tomados con los distintos experimentos.

- Persona quieta en habitación de 10 m².

Se realizaron 2 pruebas para detectar a una persona dentro de esta habitación. Los resultados para en la primera posición dentro de la habitación (Figura 5.11) fue de 18,85 segundos y además el robot tardó en recorrer toda la habitación 165 segundos.

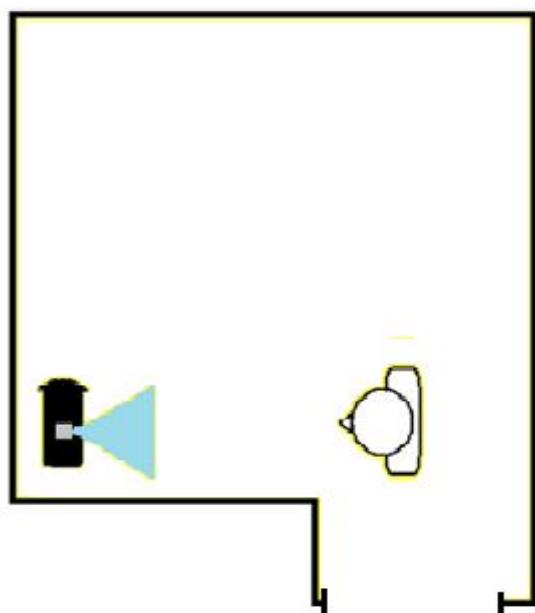


Figura 5.11: Momento detección con persona quieta en la primera posición indicada de habitación pequeña.

Para la segunda posición en la que estuvo situada la persona (Figura 5.12) se obtuvieron unos resultados de 37,35 segundos en detectarle y 182 segundos en recorrer la habitación.

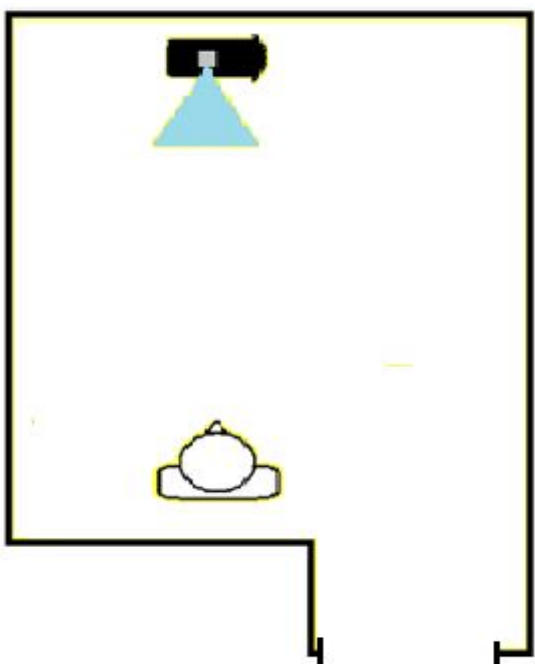


Figura 5.12: Momento detección de la persona quieta en la segunda posición indicada de habitación pequeña.

- Persona moviéndose por habitación de 10 m².

En esta prueba la persona se movió libremente por la habitación obteniendo varios resultados en cada uno de los experimentos que se realizaron en esta misma situación.

Los tiempos en detectar a esta persona por primera vez dentro de la habitación fueron de 94,85 segundos, 37,95 segundos, 53,99 segundos y 47,59 segundos.

Además el robot tardó en recorrer la habitación en cada uno de los experimentos 166 segundos, 162 segundos y 174 segundos respectivamente.

Como se puede ver estos resultados son distintos como se comentó anteriormente ya que son los tiempos en los que coincidió que esa persona miró hacia el robot.

- Persona quieta en habitación de 30 m².

En esta habitación, más grande que la anterior, se obtuvieron los siguientes resultados.

En la primera posición (Figura 5.13) en la que se situó la persona dentro de la habitación, se tardó en detectarla 69,03 segundos, tardando el robot en recorrer la habitación entera 259 segundos.

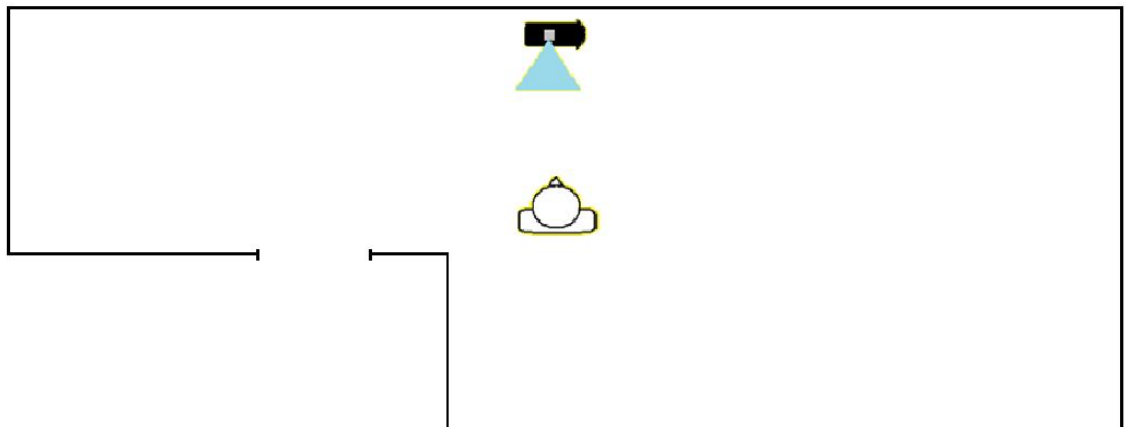


Figura 5.13: Momento detección con persona quieta en la primera posición indicada de habitación grande.

Para la segunda posición que se estudió (Figura 5.14) se tardó en detectarla 89,50 segundos y además 276 segundos en recorrer la habitación.

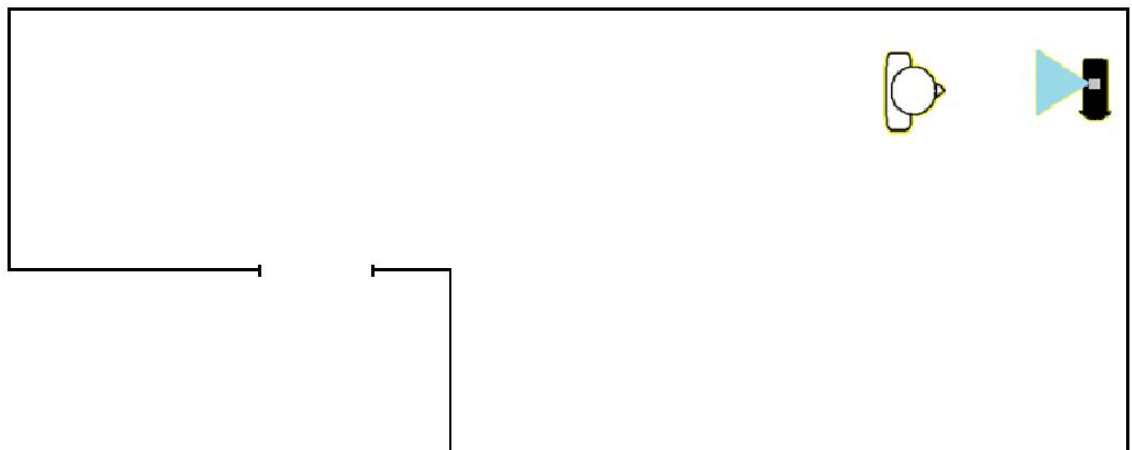


Figura 5.14: Momento detección con persona quieta en la segunda posición indicada de habitación grande.

- Persona moviéndose en habitación de 30 m².

La persona que se quería detectar para este experimento se movía libremente por la habitación de 30 m² y prácticamente era detectada cada vez que miraba hacia el robot.

Las primeras detecciones de esta persona en las diferentes pruebas realizadas fueron de 44,35 segundos, 84,60 segundos y a los 107,13 segundos.

Por otro lado, el robot móvil recorrió la habitación en tiempos de 262 segundos, 257 segundos y 273 segundos respectivamente.

5.4 Número máximo de rostros detectables.

Finalmente, en este punto, se explicaran resumidamente los resultados que se obtuvieron al comprobar el número de rostros que fue posible detectar con este sistema.

Teóricamente el sistema de detección utilizado es capaz de detectar hasta 24 rostros colocados en orden con un tamaño de 20x20 píxeles (Figura 5.15).

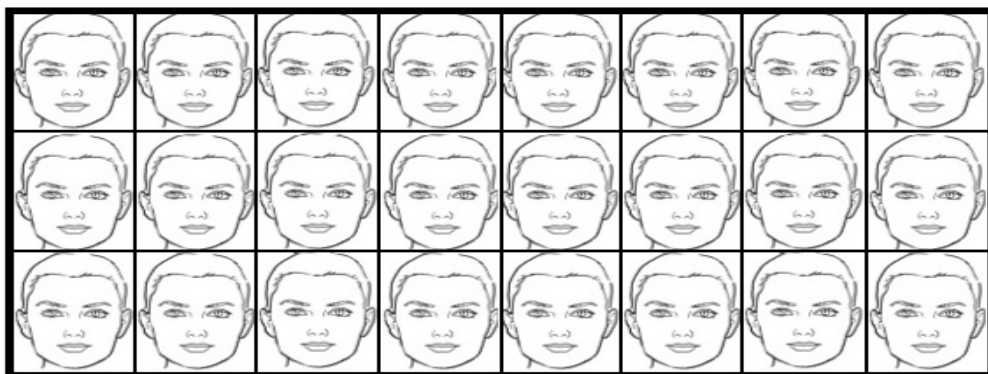
Pero esto es teóricamente ya que todos los rostros deben estar prácticamente a la misma distancia del robot y mirando todos ellos a la cámara al mismo tiempo.

Modelo de características utilizado.



20x20 Píxeles

Máximas detecciones posibles con resolución 160x60.



160x60 Píxeles

Figura 5.15: Tamaño modelo de rostros (parte superior) y tamaño del área de búsqueda de rostros de cada frame (parte inferior) con el máximo número de rostros teóricamente detectables.

En la práctica, se realizaron pruebas en varias habitaciones en las que se llegó a detectar como máximo hasta 4 personas dentro de los primeros 4 minutos en todos los casos. Al intentar detectar 5 personas no se pudieron detectar posiblemente debido a la baja resolución que se ha utilizado para los frames de la cámara.

En habitaciones pequeñas como las anteriores probadas de 10 m² el tiempo máximo tomado para detectar 4 rostros fue de 157,56 segundos.

En habitaciones grandes como la de 30 m² el tiempo máximo que se tardó en detectar 4 rostros fue de 354,53 segundos.

En ambos casos las personas eran mayores de 1,65 metros de altura y estuvieron en movimiento frente al robot en todo momento mientras estaba recorriendo la habitación.

5.5 Funcionamiento del Sistema de Detección desarrollado en otro robot.

Una vez se estudió el comportamiento del sistema de detección de personas en el robot móvil creado, se decidió comprobar que este sistema puede ser integrado en otros tipos de robots y con cámaras diferentes. Para esta comprobación se utilizó el robot Turtlebot (Figura 5.16) del laboratorio de Robótica controlado por un PC Asus Eee PC 1015PX de 1 GB de RAM, Intel Atom N570 de doble núcleo a 1,66 GHz con Sistema Operativo Ubuntu 12.04 y utilizando la versión Groovy de ROS, en el que se realizaron 3 pruebas diferentes.



Figura 5.16: Turtlebot utilizado para los experimentos.

- Turtlebot con sistema de detección de personas Raspberry Pi.

Para esta prueba se quiso demostrar que el sistema de detección de personas diseñado utilizando la Raspberry Pi puede ser un sistema independiente que puede ser montado en cualquier tipo de robot móvil sin influir en el resto del robot.

En este caso, al utilizar el Turtlebot, lo que se hizo es configurar la Raspberry Pi para mandar los topics de detecciones a la estación de trabajo del Turtlebot desde el cual se realizan todas las funciones de monitorización.

Desde esta estación de trabajo se reciben por un lado los topics provenientes del Turtlebot y por el otro las detecciones del nodo de la Raspberry Pi, por lo que se consiguen unificar todos los topics en un mismo ordenador (Figura 5.17).

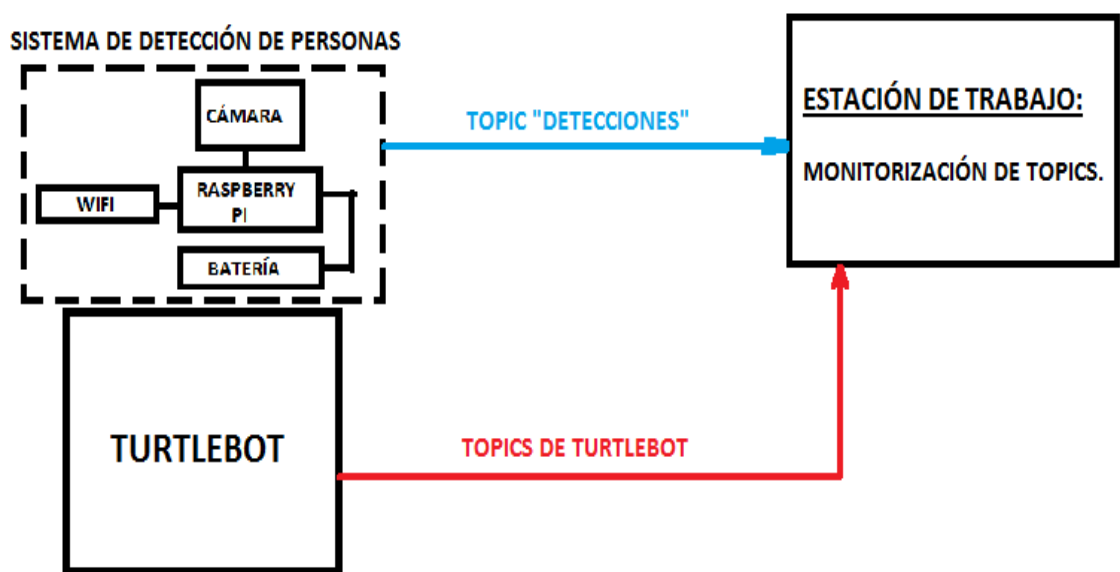


Figura 5.17: Esquema implementación del sistema de detección de personas en un robot Turtlebot.

- Turtlebot con cámara web del portátil.

En este experimento lo que se quería comprobar es que el sistema de detección de personas, diseñado en la Raspberry Pi, podía ser extraído y utilizado en cualquier otro ordenador utilizando la cámara web que tenga conectada a él, ya sea la propia webcam del portátil o una webcam conectada por USB.

En este caso el programa de detección de personas se lanzó desde el propio portátil Asus que controla el Turtlebot y se verificó que el programa creado para la detección de rostros funcionaba perfectamente utilizando otro tipo de webcam, en este caso la del portátil, de igual forma que con la cámara de la Raspberry Pi.

- Turtlebot con cámara Asus Xtion Pro.

Para este último experimento se quiso comprobar si era posible utilizar una cámara más potente como pueden ser las cámaras Kinect o, en este caso, su homóloga la Asus

Xtion Pro. El principal problema que presentaba este tipo de cámaras es que funciona a través de los drivers OpenNI, los cuales son lanzados en ROS para su funcionamiento.

En este caso, al funcionar únicamente desde ROS, se debía leer las imágenes captadas por la cámara desde los topics que mandaba el nodo de la cámara.

Para la detección de personas, se debían leer las imágenes mandadas desde el topic de imágenes RGB (/camera/rgb/image_raw) para poder utilizar el código original de detección de rostros que se utilizaba en la Raspberry Pi.

Finalmente se pudo comprobar que realizando la lectura de imágenes de topics ROS y transformando dichos topics a imágenes de OpenCV se podía utilizar el programa diseñado para la detección de rostros.

La función utilizada para transformar topics de imágenes a imágenes OpenCV es: **CvBridge().imgmsg_to_cv2(Imagen_topic, "mono8")**. Dicha función viene dada por el paquete **cv_bridge**, utilizado para transformar este tipo de topics ROS a OpenCV y, además, se transformó directamente el topic a imagen OpenCV en formato "Escala de Grises" (mono8) para, de esta forma, utilizar directamente esta imagen en la detección de rostros.

Se realizaron pruebas con esta cámara para saber si este sistema mejoraba al sistema original con la Raspberry Pi y cámara web Eye Toy. Para ello, se realizaron pruebas en la carpa del laboratorio de Robótica con iluminación artificial de flexos en el techo, ya que se consideró que serían las condiciones normales de trabajo de un robot móvil y por tanto las condiciones que se debían de probar. Con estas pruebas se consiguió detectar a un máximo de 5 personas, por lo que se consigue aumentar el número de detecciones de 4 personas, en el caso de utilizar la Raspberry Pi con la cámara Eye Toy, a 5 personas con el portátil y cámara Asus Xtion Pro. De esta forma queda demostrado que utilizando un ordenador con mayor procesamiento y mejor resolución de imágenes se consigue mejorar el sistema de detección.

6. CONCLUSIONES Y TRABAJOS FUTUROS

En este capítulo se hablará de lo que se ha conseguido lograr con este proyecto, así como la solución a los principales problemas que se plantearon al inicio para poder lograr una monitorización de la ocupación por habitaciones en entornos cerrados. Además se plantearán nuevas vías de investigación para futuros trabajos en los que se intentará dar solución o mejorar problemas que surgieron en este proyecto.

6.1 Conclusiones

Se alcanzaron solucionar los principales problemas que se plantearon al comienzo de esta memoria. Se ha conseguido poder crear desde cero un robot móvil que siga las paredes que forman el perímetro de las habitaciones y que pueda detectar rostros para poder llevar una estadística de la ocupación que hay en cada una de las habitaciones por las que pase.

Además se ha logrado realizar todo ello de forma remota. Utilizando el sistema ROS se ha podido centralizar todo ello desde un PC, ajeno al robot móvil utilizado, en el que el

PC recibe las mediciones y se encarga de realizar las estadísticas de ocupación por cada habitación.

Las estadísticas de ocupación es información que tienen muchas aplicaciones. Este tipo de información se ha pensado principalmente que puede ser utilizada por ejemplo como:

- Información semántica para identificar el tipo de estancia en la que se encuentra el robot.

Por ejemplo si hay muchas personas puede ser una sala de reuniones, de ocio, un salón, etc. En cambio si hay pocas personas puede deberse a que es un dormitorio, un despacho, un aseo, etc.

Pero además esta información se ha pensado que se puede combinar con otros tipos de información como puede ser la detección de ruido. Si juntamos la detección de personas y de ruido por habitación se puede sacar información más precisa como, por ejemplo, si en una habitación hay mucha gente y hay ruido puede asociarse a una sala de ocio, salón, etc. En cambio si en una habitación hay mucha gente y muy poco ruido puede deberse a que es un área de estudio como una biblioteca.

- Información para planificación del propio robot.

Sabiendo si una habitación está muy ocupada a ciertas horas del día se puede planificar realizar tareas en esas habitaciones en un horario que haya menos ocupación para facilitar las tareas del robot y ser más eficiente.

- Información para vigilancia o reestructuración de las estancias.

No solo puede utilizarse para planificar las tareas del robot o identificar el tipo de habitaciones sino que puede ser útil para vigilar habitaciones en horarios que no deba haber nadie o también como información para reestructurar áreas de trabajo al haber más personas en unas estancias que en otras, solicitar ayuda como un guía en salas de museos cuando sobrepase un número determinado de personas, etc.

Como se puede ver las aplicaciones que puede aportar este proyecto son infinitas.

Pueden existir mejores soluciones para realizar este proyecto y también existen dispositivos más eficaces que se podrían haber implementado en el robot para realizar las operaciones necesarias, como por ejemplo otros sensores para detectar la habitación por la que pasa, sensores de ruido para deducir el tipo de habitación, etc.

No obstante con la solución dada y dispositivos utilizados se ha conseguido realizar un robot que cumple los objetivos utilizando dispositivos reciclados que realizaban otras funciones por separado, como puede ser el caso de la cámara EyeToy y el adaptador

Wifi, y también sensores baratos como los sensores infrarrojos que detectan únicamente el color negro.

- **Colaboración con el tutor del TFG.**

La comunicación con el tutor del proyecto ha sido continua y ha servido de ayuda en casos de problemas inesperados que iban surgiendo mientras se realizaba la parte de detección de rostros al utilizar la cámara, ya que surgieron muchos problemas de Timeout debido al coste computacional que requería la detección de rostros y al procesador limitado de la Raspberry Pi pero que finalmente se pudieron solucionar y sirvió de experiencia.

- **Ingeniería aplicada.**

Para la realización de este proyecto se ha tenido que hacer uso de los conocimientos teóricos aprendidos en asignaturas de la carrera como puede ser la parte de la realización del programa de detección de rostros con las librerías de OpenCV utilizadas en una asignatura y la programación del Arduino junto con el conexionado de todos los sensores y motores.

6.2 Trabajos Futuros.

Principalmente las futuras vías de investigación en las que se podrían basar para mejorar este proyecto se centran en los 2 problemas más importantes que se han tenido al realizarlo, uno de ellos es la detección de rostros/personas y otro es la identificación de habitaciones.

- **Detección de rostros/personas:**

Este ha sido el principal problema al querer diseñar el sistema de detección de personas. Se utilizó una webcam que fuera compatible con la Raspberry Pi y que no tuviera mucho peso para que pudiera ser transportada por el robot móvil utilizado y por tanto limitó las detecciones a ser detecciones de rostro, ya que las imágenes son en 2D.

Un trabajo que se podría realizar en el futuro sería utilizar otro sistema de detección de personas para la ocupación que sea más eficaz, como por ejemplo utilizar una cámara Kinect o similar que permita la identificación por 3D y crear un algoritmo para una identificación más precisa del número de personas que hay en la habitación.

Otro trabajo que se podría hacer en base a esto es la detección exacta de las personas que hay por habitación mejorando el sistema de detección de rostros actual. Para ello,

se podría descartar los rostros ya detectados para evitar detectar a la misma persona si se va moviendo por la habitación.

- **Identificación de la habitación:**

La solución dada para saber en qué habitación se encuentra el robot en cada momento ha sido mediante la utilización de una codificación similar al del código de barras en el que se coloca un código de color negro a la entrada de cada habitación para que identifique la habitación a la que accede. Pero a pesar de ser una solución que funciona y es muy económica, se pueden realizar otras soluciones más eficientes que evitan fallos en la lectura de ese “código de barras” debido a una mala lectura que depende de la posición exacta del robot al pasar sobre ese código.

Los posibles caminos para solucionar esto serían utilizar otros sensores que sean más eficaces y más tolerantes a la posición con la que acceda el robot a la habitación, como por ejemplo utilizar tarjetas de radiofrecuencia (RFID) o un sensor de detección de color y a cada habitación se le asocia un color o una tarjeta de radiofrecuencia a la entrada.

- **Aumentar el número de monitorizaciones:**

Otro trabajo futuro que se ha pensado es el de hacer un robot que sea capaz de monitorizar y enviar a un PC o Smartphone muchas detecciones, incluida la detección de personas, como pueden ser: temperatura de las habitaciones, humedad, ruido, luminosidad, etc.

De esta forma se tendría todas las estancias monitorizadas y se podría utilizar para domotizarlas utilizando únicamente un robot móvil en vez de los sistemas domóticos actuales que son sistemas estáticos y requieren instalar los mismos sensores en todas las habitaciones.

REFERENCIAS

- [1] *Saul Stahl, "The Evolution of the Normal Distribution", Universidad Santo Tomás, Colombia, Diciembre 2008. Pag: 31.*
- [2] *Woodrow Wilson Bledsoe. "The model method in facial recognition". Technical report, Panoramic Research Inc, 1966.*
- [3] *A. J. Goldstein, L. D. Harmon, and A. B. Lesk, "Identification of Human Faces," Proc. IEEE, May 1971, Vol. 59, No. 5, 748-760.*
- [4] *L. Sirovich and M. Kirby, "Low-Dimensional Procedure for the Characterization of Human Faces," Journal of The Optical Society of America A-optics Image Science and Vision, Vol. 4, 1987.*
- [5] *J.T Jolliffe. "Principal Component Analysis". Springer Verlag. 1986.*
- [6] *T. Kohonen. "Self-organization and associative memory: 3rd edition". Springer-Verlag New York, Inc., New York, NY, USA, 1989.*
- [7] *M. Turk, A. Pentland, "Eigenfaces for recognition", Journal of cognitive neuroscience, vol. 3, No.1, pp. 71-86, 1991.*
- [8] *S. H. Jeng, Y. M. Liao, C. C. Han, M. Y. Chern, and Y. T. Liu. Facial feature detection using geometrical face model: An efficient approach. Pattern Recog., 31, 1998.*
- [9] *Lowe, D.G., "Object recognition from local scale-invariant features," Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on , vol.2, no., pp.1150,1157 vol.2, 1999.*
- [10] *Zhang, Chen, Lu and Wang, "Face Recognition Using Scale Invariant Feature Transform and Support Vector Machine", Dept. of Comput. Sci. & Technol., East China Normal Univ., Shanghai, 2008.*
- [11] *Viola P., Jones M.: Rapid Object Detection using a Boosted Cascade of Simple Features, in Computer Vision and Pattern Recognition, 2001.*
- [12] *Adrián Infiesta Aguilá, "RECONOCIMIENTO AUTOMÁTICO DE MODELOS DE AVIÓN EN ENTORNOS AEROPORTUARIOS", Facultad de Matemáticas, Universidad de Barcelona, 16 de septiembre de 2011.*
- [13] *Navneet Dalal and Bill Triggs. "Histograms of Oriented Gradients for Human Detection". IEEE Computer Society Conference. Computer Vision and Pattern Recognition, 2005.*
- [14] *L. R. Cerna, G. Cámara-Chávez, D. Menotti. "Face Detection: Histogram of Oriented Gradients and Bag of Feature Method". Computer Science Department. Federal University of Ouro Preto. 2010.*
- [15] *Rekha N, Dr. M. Z. Kurian. "Face Detection in Real Time Based on HOG". International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3 Issue 4, April 2014.*
- [16] *P.N. Belhumeur, J.P. Hespanha, and D. Kriegman. "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection". Pattern Analysis and Machine Intelligence, IEEE Transaction on, 19(7):711-720, 1997.*

- [17] *Ziad M. Hafed and Martin D. Levine. "Face recognition using the discrete cosine transform". Int. J. Comput. Vision, July 2001.*
- [18] *Laurenz Wiskott, Jean-Marc Fellous, NORbert Krüger and Christoph Von Der Malsburg. "Face recognition by elastic bunch graph matching". IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 1997.*
- [19] *FaceID Acces Control FA007 Technical Specifications. 2015.*
- [20] *"Un instituto de Badalona aplica el reconocimiento facial para evitar que los alumnos hagan novillos", Digital Magazine, 3 Oct. 2012.*
- [21] *"SceneTap Web Page". January 2015. <http://scenetap.com/>*
- [22] *"White Paper: Anonymous Video Analytics (AVA) technology and privacy", INTEL, Information and Privacy Commissioner of Ontario, Canada, April 2011.*
- [23] *"FaceCheck Verify", Cross Match Technologies, 2009.*
- [24] *"CrossMatch expands its casino presence", Homeland Security News Wire, 12 December 2006.*
- [25] *Anne-Muriel Brouet , "Emotion detectors could make driving safer", Mediacom, March 2014.*
- [26] *"NeoFace Reveal Advanced Criminal Investigative Solution Using Face Recognition Technology", NEC Corporation of America, 2013.*
- [27] *"NeoFace Watch Generate Real-time Alerts Using Highly Accurate Face Recognition Technology", NEC Corporation of America, 2013.*
- [28] *Rebecca Borison , "Chicago Police Caught A Robber With Facial Recognition Technology, And He's Off To Prison For 22 Years", BUSINESS INSIDER, JUN. 9, 2014.*
- [29] *Web Mars Science Laboratory Curiosity Rover. Enero 2015. <http://mars.nasa.gov/msl/>*
- [30] *«Curiosity, el robot más sofisticado de la NASA, llegó a Marte», INFOBAE, 6 de agosto de 2012.*
- [31] *Volatile and Organic Compositions of Sedimentary Rocks in Yellowknife Bay, Gale Crater Mars, Science Magazine Issue, Dec. 9, 2013.*
- [32] *Recent Findings from NASA's Curiosity Rover about Modern and Ancient Mars: Organics & Methane on Mars, Science Express, Dec. 16, 2014.*
- [33] *Web Page Instruments MSL, PDS Geoscience Node, Washington University in St. Louis. January 2015. <http://pds-geosciences.wustl.edu/missions/msl/>*
- [34] *"Meet the Robots of Fukushima Daiichi: A cleanup crew of automatons will go where humans fear to tread By Eliza Strickland", IEEE Spectrum, 28 Feb 2014.*
- [35] *Web iRobot PackBot 510. Enero 2015.*
<http://www.irobot.com/For-Defense-and-Security/Robots/510-PackBot.aspx#PublicSafety>
- [36] *iRobot 510 PackBot CBRNe Specifications.*

- [37] *iRobot 510 PackBot Specifications*.
- [38] *LG VR6270LVM Technical Specifications*.
- [39] *Manual Roomba serie 600, pag. 3*
- [40] *Detalles Neato Vacuum Robots*. <http://www.neatorobotics.com/es/robot-vacuum/>
- [41] *Dennis C. Pearson , "Home Automation Made Easy: Do It Yourself Know How Using UPB", Insteon, X10 and Z-Wave Brewer, 2013.*
- [42] *Raspberry Pi Hardware Specifications*. http://elinux.org/RPi_Hardware
- [43] *David Kushner, "The Making Of Arduino", IEEE Spectrum, 26 Oct 2011.*
- [44] *"AVR068: STK500 Communication Protocol", Atmel Corporation, 2006*
- [45] *Unmanned Aerial Vehicle for VideoSurveillance Using Raspberry Pi by Priyanga .M, Raja ramanan .V, Volume 3, Special Issue 3, March 2014*
- [46] *Mike Riley , "Das intelligente Haus – Heimautomation mit Arduino und Android und PC", August 31, 2012*
- [47] *Karina Gomez, Roberto Riggio, Tinku Rasheed, Daniele Miorandi, Fabrizio Granelli, "Energino: a Hardware and Software Solution for Energy Consumption Monitoring", Mayo 2012.*
- [48] *Patrick Di Justo y Emily Gertz, "Atmospheric Monitoring with Arduino", November 27, 2012.*
- [49] *Andrew K. Dennis, "Raspberry Pi Home Automation with Arduino", 2013.*
- [50] *Michael Margolis, "Make an Arduino-Controlled Robot", 2012.*
- [51] *Grimmett, Richard , "Raspberry Pi Robotic Projects", Packt Publishing Ltd, 2014.*
- [52] *Jesús Vico Serrano, "CONTROL DE UN ROBOT MÓVIL BASADO EN RASPBERRY PI Y ARDUINO", Escuela Técnica Superior de Ingeniería Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla, 2014.*
- [53] *ROS Web Page. Enero 2015. <http://www.ros.org/about-ros/>*
- [54] *Gonçalo Nuno dos Santos. "RobotTeamSim – 3D Visualization of Cooperative Mobile Robot Missions in Gazebo Virtual Enviroment". Universidad Coimbra. September 2013.*
- [55] *Jeff Huang, Cansu Erdogan, Yi Zhang, Brandon Moore, Qingzhou Luo, Aravind Sundaresan, and Grigore Rosu. "ROSRV: Runtime Verification for Robots". University of Illinois at Urbana-Champaign. 2014.*

- [56] **Kalakrishnan, Mrinal.** *"STOMP: Stochastic trajectory optimization for motion planning"*. *IEEE International Conference on Robotics and Automation (ICRA)*. 2011.
- [57] **Ciocarlie, M., Kaijen Hsiao, Leeper, A., Gossow, D.** *"Mobile manipulation through an assistive home robot"*. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.
- [58] **Nicolas Alt, Patrick Rives, Eckehard Steinbach.** *"RECONSTRUCTION OF TRANSPARENT OBJECTS IN UNSTRUCTURED SCENES WITH A DEPTH CAMERA"*. *20th IEEE International Conference on Image Processing (ICIP)*, 2013.
- [59] **Philip Parsch.** *"Simulating and Deploying Home Automation Components in Intelligent Environments"*. *Department of Electrical Engineering and Information Technology, Institute for Media Technology*. 2013.
- [60] **Arduino UNO Description Web Page.** Enero 2015.
<http://arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardUno>